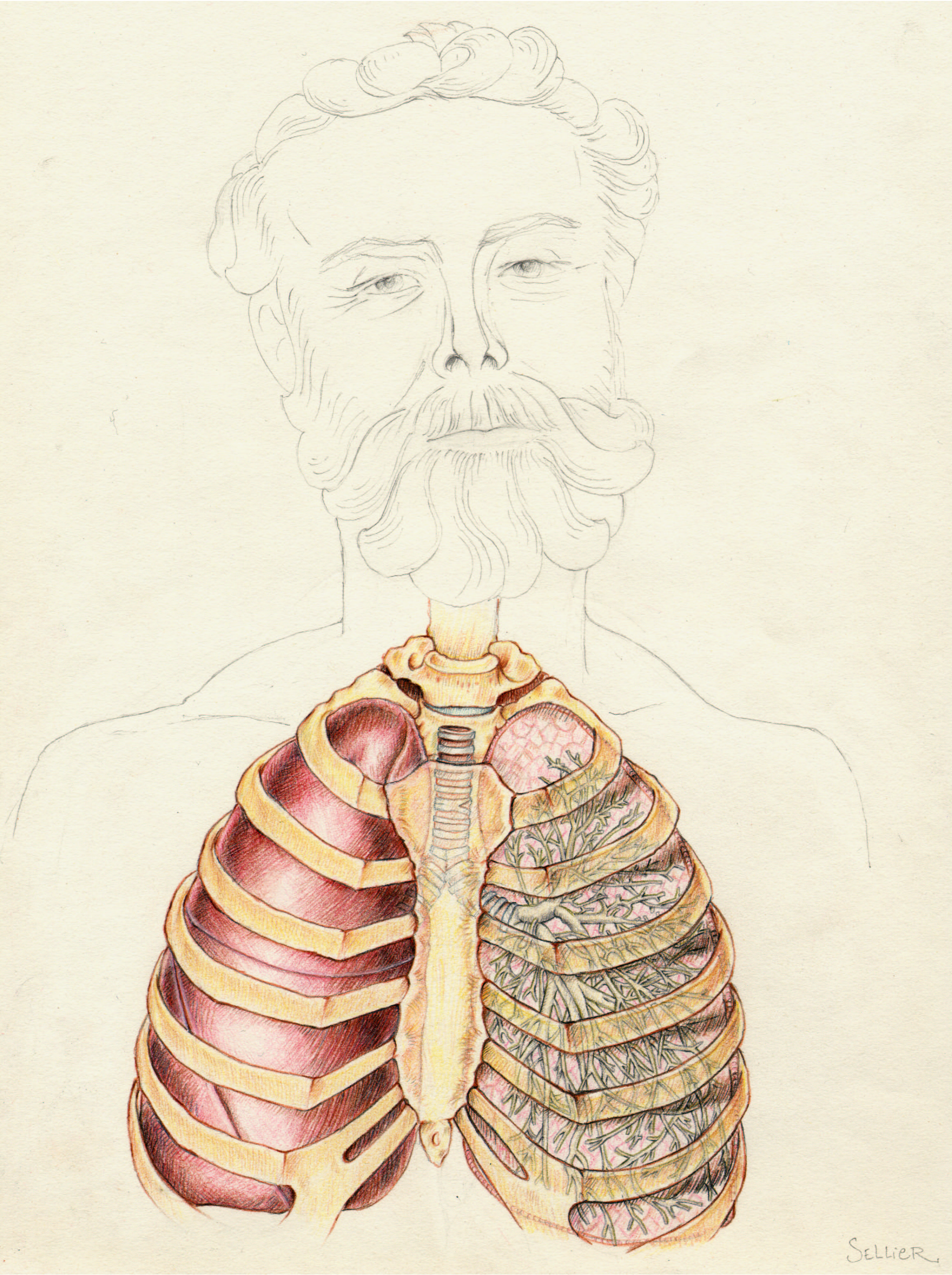
III

Visualisation et Quantification de Structures Anatomiques Arborescentes



SELLIER

Chapter 7

Outils de Visualisation et de Mesure

Résumé — Dans ce chapitre, nous présentons différents outils nécessaires pour la visualisation et la quantification de pathologies dans les objets segmentés. Comme nous l'avons déjà fait dans le cadre d'applications comme l'extraction de surfaces d'anévrismes dans des images 3DRA, ou bien la visualisation des polypes à la surface du colon, dans le chapitre 6, notre outil de segmentation de surface basé sur les ensembles de niveaux et le *Fast-Marching* permet d'extraire et de représenter des objets en 3D.
En premier lieu nous parlons brièvement des problèmes soulevés par la visualisation de surfaces implicites en 3D, et en particulier des spécificités des surfaces définies par les *Level-Sets* dans la section 7.1.
En s'appuyant sur un ensemble de trajectoires qui décrivent nos surfaces - comme le squelette dans le cas de structures arborescentes - nous développons ici des outils de mesure et d'observation des pathologies. Notamment, nous nous intéressons à la mesure du volume en section 7.1, et à la mesure de sections de nos objets segmentés en section 7.2. Ces outils seront utilisés dans toute la suite de cette partie.

Abstract — In this chapter we introduce the different necessary tools for visualization and quantification of our segmented objects. The final result of a segmentation, given by our framework as done in chapter 6 for different applications, can lead to visualization and measures on the global object.
We first briefly present the problems of visualization of an implicit surface in 3D, and more precisely the specific drawbacks of the *Level-Sets* representation in section 7.1. Assuming that we can extract a whole set of trajectories in a tree-shaped object, we present the different tools that will measure the pathologies, on the basis of those trajectories. Important measurements include: volume measurements, as explained in section 7.1, and objects cross-section measurements, as detailed in section 7.2. Those tools are useful for the framework developed in the following chapters.

7.1 Visualization of 3D segmentation

In the domain of medical image analysis, the segmentation tools we developed are essentially interesting when applied to 3D images. This is the reason why the implementation we build is designed for this image dimensionality, and that our visualization efforts were mainly directed to 3D techniques.

In this section the reader will first find a short presentation of the basic notions of virtual reality needed to understand the content of our work. They are grouped together in the first subsection, which can be skipped by the readers who are already familiar with them.

7.1.1 Virtual reality notions

Classically, the basic techniques for computer graphics of virtual reality rely on the computation of *renderings* of virtual 3D *scenes*. A scene is composed of virtual *actors*, *lights* and a *camera*.

What are actors ?

The term *actor* covers everything that might be seen when properly enlighten. For instance, in a virtual reality model of a house, each piece of furniture would be modeled by a specific actor, and so would be the floors, walls, stairs, etc.

Traditionally, the shape of a 3D actor is explicitly modeled by a set of graphic primitives: points, lines and surface patches. In recent and advanced models, the surface of an actor is sometimes modeled using implicit functions.

According to the complexity of the modelization, the rendered appearance of the surface of an actor can depend on many and various parameters:

- the position and orientation of the camera relatively to the actor surface;
- the properties of the surface which are taken into account by the *illumination model*;
- the positions, orientations, colors and attenuation factors of the lights, which can be at finite distance (punctual lights) or infinite distance;
- the positions and orientations of the other actors which may cause occlusions, projected shades, or even reflect light sources in advanced models.

What is an illumination model ?

The illumination model is the set of equations used to compute the color and brightness of a point on the surface of an actor according to:

- the angles of incidence, intensities and colors of the incident rays of light;
- the modeled properties of the surface;
- the angle of the departing ray of light.

The surface properties usually include colors, an opacity factor, reflectance, a specular power parameter, etc...

In addition to the illumination model, a *shading* model can be used to avoid the faceted aspect of polygonal surfaces. The most popular shading models are Gouraud and Phong shadings, although Phong shading is rarely used because of its computational cost.

What is the exact role of the camera ?

The camera plays the same role as in the making of a movie: the rays of light which encounter the objective determine the rendered (i.e. virtually acquired) 2D image. The usual parameters of a camera are its position, orientation, and two of the following parameters: view angle, focal distance and image size. The rendered 2D image is a projection of the illuminated actors in the focal plane of the camera.

Object-based and image-based rendering

The construction of the 2D image acquired by the camera may be *object-based* or *image-based*. In the case of object-based rendering, the actors are rendered one by one by applying the illumination model and the projection equations to the graphic primitives they are composed of. The occlusions are generally dealt with using a so-called *Z-buffering* technique: the final image is the result of the superposition of layers which correspond to different depths (Z-coordinate) in the scene. The points that are the closest to the camera are visible, others are more or less occluded according to the opacity of the points that are in front of them.

Object-based rendering is not a recent technique, but it is fast, rather simple, and can be accelerated by specialized hardware devices. For example, OpenGL hardware implementations make interactive renderings of simple scenes possible even on a low end PC. The main drawbacks of object-based rendering are that photo-realistic images are difficult to achieve, especially in the case of complex scenes, and that multiple reflections are usually not taken into account. Moreover the actors have to be explicitly represented using graphic primitives.

In the case of image-based rendering (or *ray-tracing*) the color and brightness of each point of the rendered image is computed by tracing a ray starting from this point. The illumination model is invoqued when the ray hits an actor, and reflections on several actors are even possible before reaching a light source. In the most advanced computer graphics software products based on ray-tracing, the actors can also have implicit representations.

The images produced by ray-tracing can be of very high quality, but the major drawback of image-based rendering is the computation cost related to the calculation of the rays.

7.1.2 Visualization of a level-set

Visualizing a level set is nothing more than visualizing an iso-surface in 3D, or an iso-contour in 2D. More generally the hypersurface which needs to be visualized is the zero-level set of $\phi(\cdot, t)$, where t is fixed, which is a function defined over the image domain $\Omega \subset \mathbb{R}^3$ in our applications. In figure 7.1-left, the surface of a sphere is implicitly defined by the signed distance to itself.

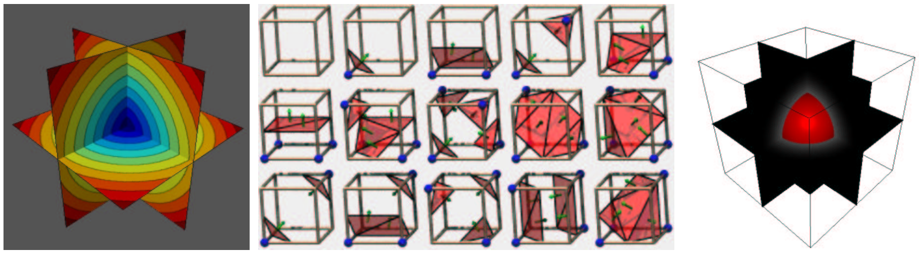


Figure 7.1. The marching cubes algorithm: Left image represents the iso-values of the signed distance to a sphere in 3D; middle image represents the different configuration encountered by the *Marching-Cubes* ; right image is a smooth surface rendering of the triangles that approximate the implicitly defined sphere of figure 7.1-left given by the algorithm.

The 3D visualization of the zero level-set surface by object-based rendering algorithms cannot be done directly. An explicit representation of the surface by polygonal graphic primitives has to be computed first.

Several approaches are possible for the computation of a polygonal approximation to an iso-surface. The most popular of all is certainly the *Marching-Cubes* (see [104]), which computes a triangulated surface. In each cube formed by eight contiguous voxel centers, the values of the implicit function at the vertices of the cube are compared to the specified iso-value. The possible configurations are classified (see figure 7.1-middle), and a look-up table is used to quickly give a triangulated approximation of the intersection of the iso-surface with the currently examined cube. All the cubes are examined one by one in a raster-scan "marching" fashion, in opposition to the algorithms which try to "track" the iso-surface.

But sometimes the *Marching-Cubes* algorithm generates triangle sets containing holes, due to ambiguous cases. Many authors have proposed solutions, for example the marching tetrahedra algorithm in [166].

However, we chose the *Marching-Cubes* for reasons of accuracy, reliability, and (above all) simplicity of use since efficient implementations of it are available. It provides an accurate triangulated surface whose precision leads to high-quality renderings, like in the endoscopic images shown in figure 7.2.

7.1.3 Problem with the *Marching-Cubes*

A classical evolution equation defined by $\frac{\partial \phi}{\partial t} + V \, . \, \nabla \phi = 0$ makes no distinction between the level sets of ϕ. They are all attracted by the same asymptotic hypersurface provided that they are sufficiently "close" to it. As a result, ϕ gets very steep in its vicinity, which causes the *Marching-Cubes* to give poor and aliased results.

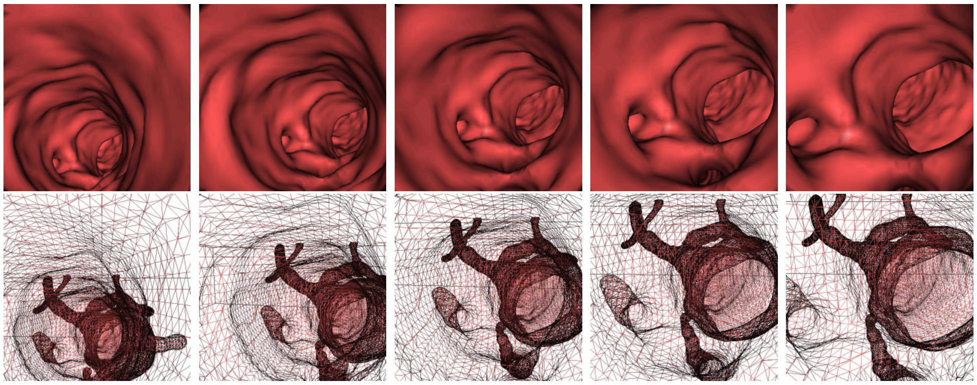


Figure 7.2. Surface rendering in the aorta: First row shows frames of an endoscopic movie in an aorta MR dataset. Second row is the wire-frame version of this movie, given by the *Marching-Cubes* where we can see the whole anatomical object and its several branches by transparency

In fact, the level-sets do not remain a distance function in many cases (an exception is a constant advection flow, for example see figure 4.6). This property at initialization, is lost after several iterations. Figure 7.3 shows two examples: the first one follows a balloon forces, which is positive inside a circle, and negative outside; the second one is a flow composed of a positive balloon force and a boundary based force, which stops the level sets of ϕ.

7.1.4 Restoring the distance function

In conclusion, the solution to the classical Hamilton-Jacobi evolution equation proposed in [135] is not a distance function. But this property is the hypothesis of several numerical techniques to accelerate convergence, like the fast geodesic active contours proposed in [65] and [67]. Moreover, the practical application of the level-set method is plagued with such questions as: when do we have to "reinitialize" the distance function? How do we reinitialize" the distance function. In [163], the author suggests that when the zero-level set evolves in the vicinity of the borders of the narrow-band, the distance to the zero-level set must be re-initialized. For the authors of [68, 69], this problem reveals a disagreement between the theory and its implementation, the authors propose an alternative to the use of Hamilton-Jacobi evolution equation which eliminates this contradiction. In order to reach this goal, they look for a function $B : \mathbb{R}^3 \times \mathbb{R}^+ \rightarrow \mathbb{R}$ such that $\frac{\partial \phi}{\partial t} = B$ and which satisfies the two constraints

- ϕ is a distance function
- $\frac{\partial \phi}{\partial t} = \beta \mathcal{N}$ where β is the velocity, and $\mathcal{N}$ the inward unit normal.

Those constraints lead to the new relation $\nabla \phi \cdot \nabla B = 0$. This efficient method increases the computing cost.

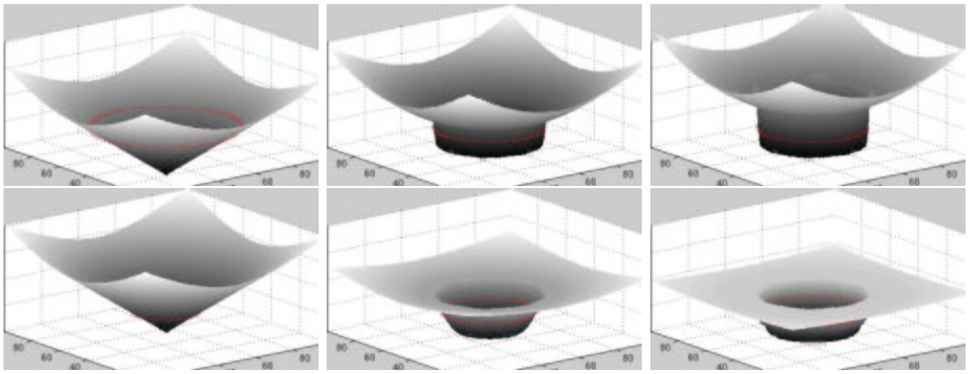


Figure 7.3. Loosing the distance function when converging: First row shows consecutive iterations of the level-sets of a geodesic active contour that minimizes the distance to a circle; second row shows consecutive iterations of the level-sets of a geodesic active contour that inflates according to a balloon force with boundary based forces on the same circle.

Following the author of [84], we include the a restoration force in the Hamilton-Jacobi flows, which not only ensures the evolution of $\phi(0,t)$ given by equation (4.6), but also prevents $\phi(\cdot,t)$ from getting too steep in the vicinity of $\phi(0,t)$. This new partial differential equation is given by:

$$\frac{\partial \phi}{\partial t} + V \,.\, \nabla\phi = \mu \,.\, \mathrm{sgn}_\theta(\phi(x,t)) \,.\, (1 - \|\nabla\phi\|) \tag{7.1}$$

where V is the flow defined by equation (4.6) and where the modified signed function sgn_θ is defined by:

$$\mathrm{sgn}_\theta(y) = \begin{cases} -1 & \text{if} \quad y < -\theta \\ \frac{y}{\theta} & \text{if} \quad -\theta \le y \le \theta \\ 1 & \text{if} \quad \theta < y \end{cases}$$

The new differential operator introduced in equation (7.1) is inspired from the distance function restoration operator used in [170]. The modification of the sign function avoids the apparition of oscillations during the numerical approximation of equation (7.1), without having to introduce numerical flux or slope bounds. Otherwise, as signaled in [84, page 56], these oscillations are responsible for short but annoying displacements of the zero-level set of $\phi(\cdot,t)$. And the author of [84] proposes to use another scheme, originally presented in [159], which inflates and deflates successively the level-set in order to extract the distance to the zero level-set, without displacing it. We choose not to add another bunch of computations to our method, and decide to use method of [170].

The parameter denoted by θ can be set to a fixed value (we used $\theta = 10$ in our experiments). The parameter μ defines the weight of the newly introduced differential operator, and has to be adapted according to the other forces parameters. If μ is too small, then $\phi(\cdot,t)$ is likely to get too steep for the *Marching-Cubes* to give good results. But too high values of μ will increase the global CFL number, and thus cause the convergence of ϕ to be slower. In practice, it is not difficult to find a good value for μ.

The use of this new equation (7.1) is illustrated in figure 7.4, where the flow drives the zero level-sets to a sphere.

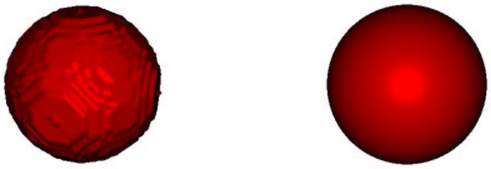


Figure 7.4. Aliasing when converging: Left image is the surface extracted at convergence when the level set matches the surface; Right image is the same result including a force to restore the distance function.

7.1.5 Volume versus Surface Rendering

Volume rendering is an advanced image-based visualization technique based on the integration of a transfer function along rays cast in a dense volume (i.e. a 3D image). The transfer function is generally based on the intensity and gradient of the image, and gives an opacity value for each voxel of the image. Surface (see figure 7.5-left) versus Volume (see figure 7.5-right) rendering is still an open question, and the choice between those two methods depends on the application.

With the shape extraction techniques we use, surface rendering has several advantages:

1. with the segmentation framework we have developed, the visualization of the anatomical object with surface based rendering does not need any input, any interaction (unless the color of the surface can be considered as an important parameter);

2. parameterization-free means robustness. Volume-based rendering relies on the critical choice of a suitable transfer function. Surface-based rendering is the direct representation of the surface extracted by the segmentation whereas the volume-based rendering relies on the user perception of the dataset;

3. Surface rendering is fast: when the triangulation has been extracted with the *Marching-Cubes* , endoscopic fly-through, like in figure 7.2 are generated in real-time, and OpenGL hardware implementations, now available on any low end PC, accelerate the computations. The computational cost of volume rendering

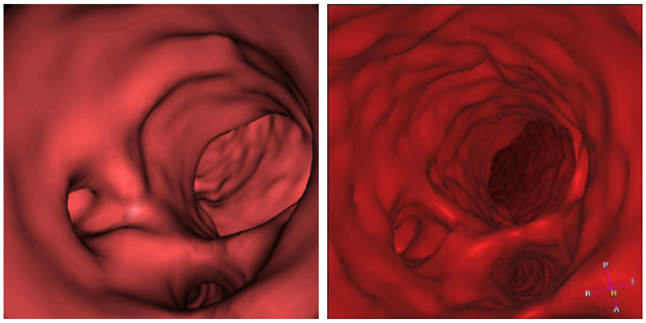


Figure 7.5. Comparing surface and volume rendering in the aorta: Left image is a surface rendered view generated with the *Marching-Cubes* on the final segmentation obtained; right image is a threshold based volume rendering view at the same location in the dataset.

is very high, and special hardware devices that might overcome this lack of performance are still under development.

Moreover, several artifacts may occur when using *Volume Rendering* on volume data (among them, aliasing, stair-casing and slicing, see [146]).

For all those reasons, we used the surface-based rendering for visualization, as well for inspection of results, as for endoscopic viewings. Notice that if surface-based rendering is parameter-free, it critically relies on the result of the segmentation.

7.2 Measurement Tools

The main target of our path and shape extraction framework is to measure pathologies in tube-shaped objects, like aneurysms in brain vessels, and polyps in the colon. We detail in this section the different tools used for quantification of those pathologies, that are characterized by their sections and volumes. Extracting the shapes of our objects, with the *Marching-Cubes* [104], we use a consequence of the Gauss theorem, discretized on the vertices of the triangulation obtained.

7.2.1 Gauss Theorem

As classically [9], volume and section measurements are based on Gauss theorem:

Theorem 7.1 (Gauss) *Let* Ω *be a subset of* $\mathbb{R}^d$*, let its boundary* Σ *be a closed surface, and* U *a differentiable vector field, then:*

$$\int_{\Omega} div\, U \, dx = \oint_{\Sigma} U . N \, d\sigma$$

where N *is the outward normal to* Σ*.*

A consequence of Gauss theorem is that the volume $\mathcal{V}(\Omega)$ of Ω can be simplified as an integral over the boundary Γ

$$\mathcal{V}(\Omega) = \int_{\Omega} dx = \frac{1}{3} \int_{\Omega} \operatorname{div}(x)\, dx = \frac{1}{3} \oint_{\Gamma} x.\, N\, d\sigma. \tag{7.2}$$

7.2.2 Volume Measurement

We assume that a 3D tubular structure has been segmented with a level set method and that $\phi(\cdot, t)$ is known at convergence and denoted by $\tilde{\phi}$. We also assume that centered paths have been computed in the tubular structure.

The volume to be measured is defined by the user who chooses a path and specifies two points p_1 and p_2 on this path. The computed volume is the volume of the interior region of the tubular structure limited by the two plane section S_1 and S_2 associated to (p_1, Π_1) and (p_2, Π_2) and defined by $S_i = \Pi_i \cap \tilde{\phi}^{-1}(\mathbb{R}^-)$ $i = 1, 2$. Here is a step-by-step summary of our algorithm, which is illustrated by figure 7.6.

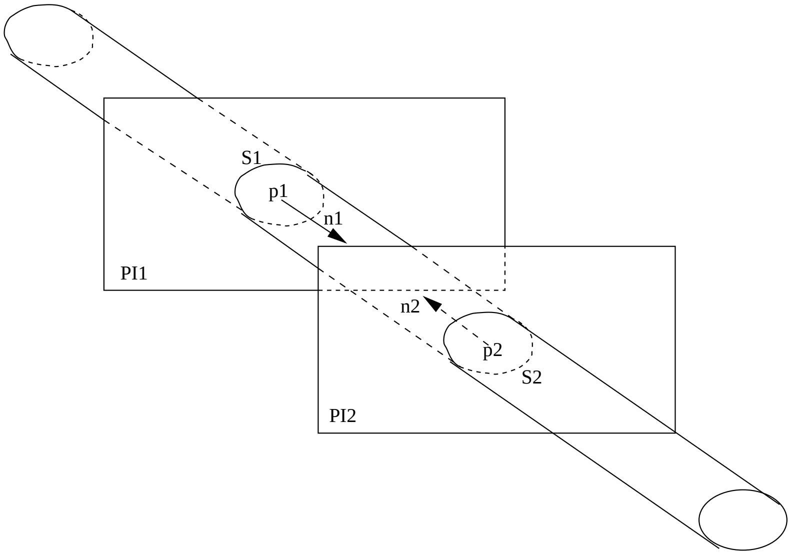


Figure 7.6. Volume measurement diagram.

- we compute tangent vectors to the path at p_1 and p_2, which are the normal vectors $\overrightarrow{n_1}$ and $\overrightarrow{n_2}$ to the plane sections S_1 and S_2;
- the equations $(p_1, \overrightarrow{n_1})$ and $(p_2, \overrightarrow{n_2})$ of the plane sections S_1 and S_2 are considered;
- the region of interest is actually the intersection of three subsets of $\mathbb{R}^3$, which are $\tilde{\phi}^{-1}(\mathbb{R}^-)$ and two half-spaces limited by the plane sections;
- we deduce the signed distance functions Ψ_1 and Ψ_2 to the two half-spaces Π_1 and Π_2 from the equations of the plane sections

- Considering that the shape of the object of interest can be complex, and lead to problems of intersection between planes Π_1 and Π_2 (see figure 7.8), we define a function Ψ the following way
    1. it is initialized with $\Psi(x, y, z) = \sqrt{3}, \forall(x, y, z)$ in the image domain;
    2. starting from the path point p_1 (respectively p_2), we apply a region growing algorithm, that labels only the voxels v which have $|\Psi_1(v)| < \sqrt{3}$, (respectively $|\Psi_2(v)| < \sqrt{3}$) and for those voxels, we set $\Psi(v) = \Psi_1(v)$ (respectively $\Psi(v) = \Psi_2(v)$);
    3. starting from any path point p not labeled between p_1 and p_2, we apply a connectivity filter that visits only the voxels where $\tilde{\phi}(v) < \sqrt{3}$ and that are not already visited by the first connectivity filter; and for each voxel visited we set $\Psi(v) = \tilde{\phi}(v)$; it enables to avoid including the interior of undesired parallel structures in the region of interest;
- the region of interest is equal to $\Psi^{-1}(\mathbb{R}^-)$, and a polygonal approximation of its boundary is computed by extracting the zero-level set of Ψ;
- the volume of the region of interest is computed using the following decomposition of equation (7.2) on the polygons of the extracted surface:

$$\mathcal{V}(\Psi^{-1}(\mathbb{R}^-)) = \frac{1}{3} \sum_{\mathrm{i}} \mathrm{g_i}.\,\mathrm{N_i}\,\sigma(\mathrm{P_i})$$

  where g_i, N_i and $\sigma(P_i)$ respectively denote the center of gravity, the outward normal and the surface of the polygon P_i.

The overall computation times are very short (less than 3 seconds for a $256 \times 256 \times 60$ image on a SunBlade 100), and the results on basic geometric primitives are excellent in terms of accuracy.

7.2.3 Example of volume measurement: an aneurysm

In this case, shown in figure 7.7, where the problem studied is the cerebral aneurysm of figure 6.4, the measurement of the aneurysm volume is done using one trajectory extracted inside a mask defined by the segmentation obtained in figure6.6. Taking two positions along the trajectory, we can easily define a volume of interest that contains the aneurysm. The volume shown in figure 7.7-right is not restricted to the aneurysm itself, and contain the surrounding vessel. But a good approximation can be given, by subtracting an approximate vessel volume, using the surfaces of the sections S_1 and S_2. Advantage of using our connectivity algorithm to obtain $\Psi(\mathbb{R}^-)$ instead of taking the region delimited by $\tilde{\phi}^{-1}(\mathbb{R}^-)$ and the two half space Π_1 and Π_2 determined by the distance functions Ψ_1, Ψ_2 is illustrated by figure 7.8 on the same dataset.

7.2.4 Section Measurement

We can also apply equation (7.2) in 2D to evaluate the surface limited by a closed planar curve. In order to illustrate this method, we show its application to a phantom dataset.

The data, shown in figure 7.9-left is the acquisition of a cube of Perspex (a type of plastic) with an aluminum rod in it, inside a dead human head. It was acquired

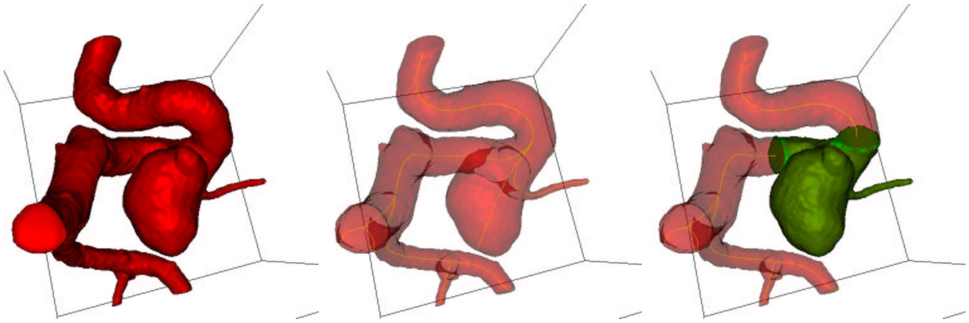


Figure 7.7. Measuring the volume of an aneurysm: The dataset used for this segmentation is shown in figure 6.4. Left image is the segmented object obtained in figure 6.6 by combining *Fast-Marching* and *Level-Sets* methods; middle image shows the multiple paths extracted; right image shows a sub-volume of the aneurysm which has been isolated.

with the Philips Integris **3D-RA** system. A **MIP** view in figure 7.9-right enables to see the variable section of the aluminum rod.

Following the results of chapter 6, we first segment the phantom with the *Fast-Marching* algorithm, starting from one point at the top of the aluminum rod. Computing the Euclidean path length while propagating, as detailed in section 2.2.3, it is very easy to extract the largest centered path, using the method described in section 2.3, with the thresholded distance $\tilde{\mathcal{D}}$ to the object borders. This path extracted is visible in figure 7.10-middle, by transparency. In a few iterations, the *Level-Sets* algorithm, with region-based forces, gives the result shown in figure 7.10-left.

In the experimental tool we built, the user specifies a particular path and obtains the section of the tubular structure according to the length of the path. The path is supposed to have a discrete representation, i.e. is represented by a list of points. Here we give a summary of the performed calculations for each point of the path:

- the normal of the section plane is computed using an approximation of the tangent vector to the path;
- an orthonormal base of the section plane is deduced;
- a rectilinear 2D grid, centered on the current path point, is defined on the section plane;
- at the center of each cell of the grid, the value of $\tilde{\phi}$ is computed by interpolation;
- an adequate algorithm is used (we used the Marching Squares) to compute an approximation of the zero iso-contour in the 2D grid;
- the surface enclosed in the resulting polygonal line, which in our example is drawn on the surface in figure 7.10-right, is computed thanks to a decomposition of equation (7.2) on the polygonal line.

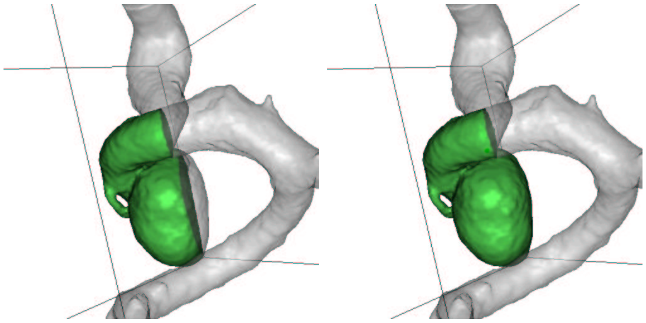


Figure 7.8. Advantage of the connectivity algorithm: left image shows the result of computing the intersection of $\tilde{\phi}^{-1}(\mathbb{R}^-)$, $\Psi_1^{-1}(\mathbb{R}^-)$, $\Psi_2^{-1}(\mathbb{R}^-)$. Right image is the representation of $\Psi(\mathbb{R}^-)$ superimposed on the segmentation along the same trajectory, between the same extremities.

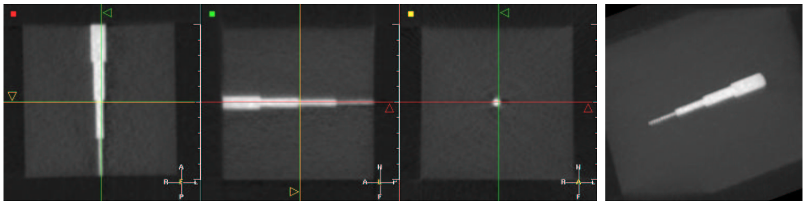


Figure 7.9. 3D-RA Phantom: On the left image are shown three orthogonal views of the perspex cube acquired with a **3D-RA** system; right image is a **MIP** view of this data-set.

Like in the case of volume measurements, the computation times are very short, and the algorithm gives very accurate measurements of basic geometric primitives. Concerning the phantom problem, we have computed this section at each path point (see figure 7.11). Figure 7.11 shows the measures done along the path displayed in figure 7.10-right. On the graphic, we have displayed the several real dimensions of the aluminum cylinders, and we have also displayed the interval of deviation of 2% that was indicated by a study on the accuracy of the calibration, the distortion correction, and the reconstruction of the **3D-RA** system [83]. The section measurements of the segmented object show that our method gives results which lie in those intervals, when the radius is more than one millimeter.

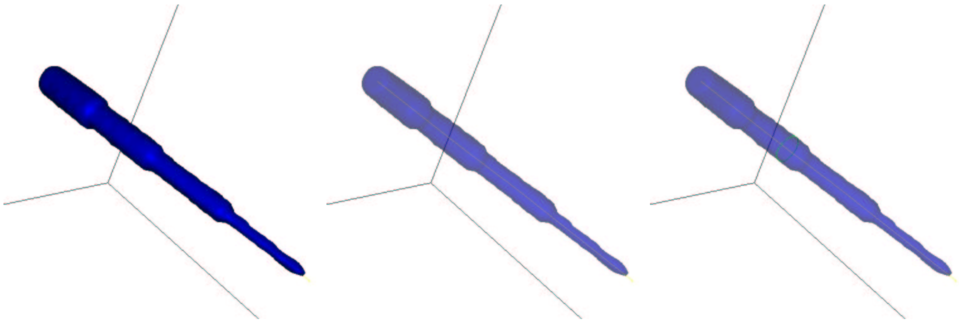


Figure 7.10. Segmentation result on the 3D-RA Phantom of figure 7.9: Left image is the segmented object with the combination of *Fast-Marching* and *Level-Sets* methods; middle image is the same object with opacity $< 1.$ and the path extracted; right image shows the intersection of the phantom surface with the section plan for measurements.

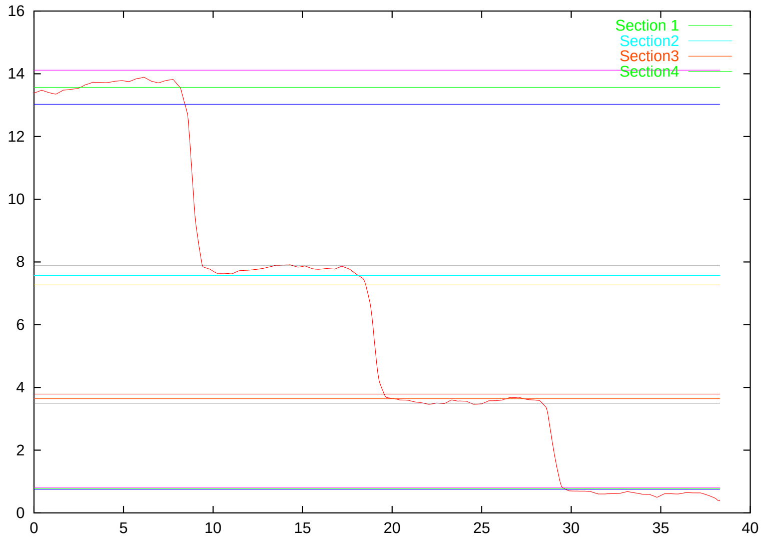


Figure 7.11. Section measurements along the segmented phantom of figure 7.10: It represents the values of the section across the trajectory extracted, with the deviation of 2% superimposed.

Chapter 8

Extraction de Structures Arborescentes

Résumé — Dans la partie précédente nous avons présenté un algorithme mettant en oeuvre une collaboration entre le *Fast-Marching* et les *Level-Sets* pour la segmentation. Dans ce chapitre, nous souhaitons présenter une application de cette collaboration spécifiquement dédiée aux structures arborescentes du type arbre vasculaire.
Tout d'abord nous montrons comment le *Fast-Marching* permet de fournir une présegmentation rapide et précise pour les structures arborescentes dans la section 8.3.
Nous utilisons ensuite les *Level-Sets* de la même manière que dans la section 5.4 de la partie précédente.
Finalement nous montrons comment le *Fast-Marching*, déjà utilisé pour l'extraction de trajectoires dans la partie I, permet aussi d'extraire plusieurs trajectoires et de remonter à l'information d'arbre ou de squelette d'un objet tubulaire avec embranchements multiples.

Abstract — In the previous part I, we detailed an algorithm using *Fast-Marching* and *Level-Sets* in a collaborative manner for object segmentation. In this chapter, we introduce an application of this collaboration specifically adapted to tree anatomical structures, like vascular or arterial tree. First of all, we demonstrate in section 8.3 the ability to build a fast and accurate pre-segmentation for those tree structures using a dedicated *Fast-Marching* algorithm.
We further apply the *Level-Sets*, as in section 5.4, for converging to a more accurate solution.
Finally, we show in section 8.4 how the *Fast-Marching* ability to extract trajectories, as used in part I, can be extended to the simultaneous extraction of multiple trajectories, and to obtain the underlying tree structure of a tubular anatomical shape with several branches.

8.1 Introduction

In the first part of the thesis, we have implemented several techniques to extract a trajectory inside a tubular structure. We have shown application of this fast minimal path-extraction process to automatic and interactive methods to extract lineic structures in images. In the second part of the thesis, we have combined fast and accurate methods for shape extraction, using the same kind of grey-weighted distance transform algorithms. We have proved the ability of those techniques to extract surfaces, and to emphasize pathologies, in several applications. In the last part of the thesis, we now want to integrate the path and surface extraction algorithms, in order to present an accurate global framework for the segmentation, the visualization, and the quantification, of anatomical objects. In the previous chapter, we have detailed the algorithmic techniques to obtain representations and measures of our anatomical objects, based on extracted primitives of our objects like shapes and skeletons. In this chapter, we will present the basic framework, and extend its possibilities to the detection of tree-like structures, and their corresponding set of multiple trajectories, in order to enhance measures and visualization of pathologies of any tube-shaped object.

This chapter will be illustrated by applications of the algorithms presented on the segmentation and quantification of vessels in contrast-enhanced 3D medical images.

8.2 Motivation

We have seen in part II a method to use front competition for image segmentation. This process involved to visit the whole image domain, and was not tuned for a particular category of objects. Moreover, in huge images, as multi-slice CT scanners (see application to lungs images in chapter 9.2), the visit of the whole image cannot be done in interactive time.

8.2.1 Tree extraction

In this chapter, we are focusing on the extraction of thin tubular structures. Our algorithms can be dedicated to this particular category of tube-shaped objects. If the propagation of a front could be restricted to the part of the image occupied by those structures, the computing time could be divided by almost 5, since vessels in a typical MR-Angiography image do not exceed 10% of the whole volume.

In chapter 2, we have developed an algorithm that can be the basis of this kind of tubular shape extraction object: a technique to evolve a front inside an object of interest and compute at the same time the Euclidean distance to the start point. It was used to reduce the user interaction to locating only one extremity of the path inside a tubular structure. This Euclidean distance can be used to stop the front propagation inside the desired object. If we have precise knowledge of its length, we can decide to stop when this given length has been reached in the expression of the Euclidean path length computation, as explained in section 2.2.3. The result of this technique is shown in figure 8.1.

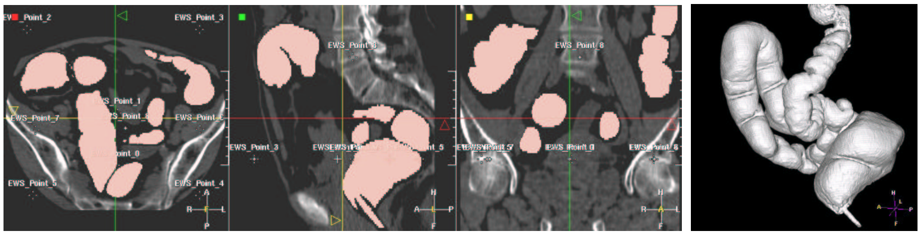


Figure 8.1. Segmenting the colon volume with simple front propagation: as in the virtual endoscopy facility, the user locate a starting point at one particular recognizable part of the colon, then a front is propagated from this seed point until a maximum path length is reached. Left image represents the datasets where the intersection with the segmented object is visible in pink. Right image is the 3D volume rendering of the final segmentation.

However, classical segmentation problems do not provide an excellent contrast like the air-filled colon on a CT scanner, and the propagation cannot stick to the object walls, as it is shown in figure 3.6. For example, if we apply the same kind of propagation in the dataset shown in figure 3.12 for the endoscopy application in chapter 3, the corresponding wave propagation looks like figure 8.2. The front floods outside the object and cannot be used as an initialization step for a more complex segmentation, like the combination of the *Fast-Marching* and the *Level-Sets* which was presented in the previous part.

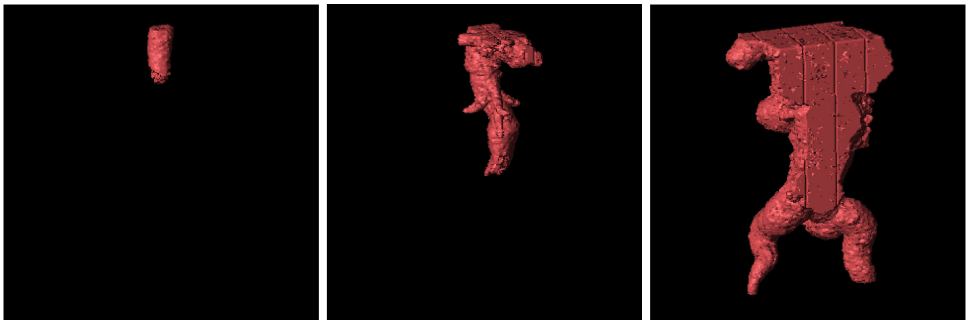


Figure 8.2. Wave propagation inside the aorta MR dataset: These three images represent different steps of the propagation inside the aorta MR dataset using *Eikonal equation* equation with a potential similar to the one defined for the endoscopy application (a simple function of the grey levels either linear or non-linear).

In the following section, we will present a new algorithm, based on the *Fast-Marching* and dedicated to a *quick and dirty* segmentation of the tree structures in 3D medical images.

8.3 Design of an adequate initialization algorithm

We have shown the possibility to provide efficiently an initialization for more complicated methods in the previous part of the thesis. Setting up a framework for the visualization and the quantification of thin tubular structures, based on the same combination of the *Fast-Marching* and the *Level-Sets*, we show in this section how the previous initialization step, which is not tuned for this kind of thin and long objects, can be specifically optimized for this target.

8.3.1 Propagation Freezing for Thin Structures

Freezing a voxel during front propagation is to consider that it has reached the boundary of the structure. When the front propagates in a thin structure, there is only a small part of the front, which we could call the "head" of the front, that really moves. Most of the front is located close to the boundary of the structure and moves very slowly. For example voxels that are close to the starting point, the "tail" of the front, are moving very slowly. However, since the structure may be very long, in order for the "head" voxels to reach the end of the structure, the "tail" voxels may flow out of the boundary since their speed is always positive. This is illustrated in the example of figure 3.12. If we apply fast marching in the dataset shown in figure 3.12-top with a potential based on the gray level with contrast enhancement the corresponding wave propagation looks like figure 8.2. The front floods outside the object and does not give a good segmentation.

For these reasons, it is of no use to make some voxels participate in the computation of the arrival time in *Eikonal equation* by setting their speed to zero, which we call *Freezing*. First step is to design the appropriate criterion for selecting voxels of the front which needs *Freezing*.

Concerning the application to the tree tracking, the several improvements brought by this method are

- to accelerate the computations, by visiting a very small number of voxels during propagation;
- to enable the segmentation of thin tubular structures;
- therefore enabling the centering inside those tubular structures.

First step is do design the appropriate criterion for *Freezing* voxels of the front. We illustrate this *Freezing* principle on a synthetic branching structure in 2D.

Synthetic test problem

A synthetic example of a tree structure is shown in figure 8.3. In this case, setting an initial seed point at the hierarchy, we would like to extract in a very fast process the multiple branches of the structures, and its corresponding skeletonization, in a single process. Figure 8.3 shows the result of the classical front propagation technique with the *Fast-Marching* coupled with a maximum Euclidean path length stopping criterion. The action map displayed clearly indicates that the domain visited is a whole

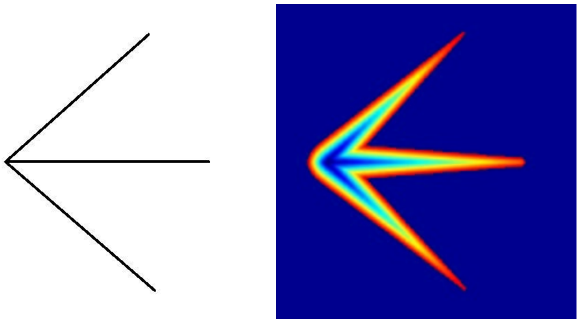


Figure 8.3. Synthetic test problem: The left image is the medium where a front has been propagated, starting at the root of the three branches, and stopping when a maximum distance criterion of 300, computed according to method described in section 2.2.3, has been reached; Right image is the corresponding action map.

"blob-like" structure where the underlying tubular shape is somehow lost. Therefore, tracking a minimal path from the regional maxima of the action map will not lead for sure to paths that stay inside the object of interest. It emphasizes the little use of this method, without a clear constraint on the domain of points visited.

Using Time for Freezing

The heuristic presented in this section is to discriminate the points of the front that are spending a long time in the propagating front, i.e. points that are visited but whose action is not *frozen*, in the sense defined in table 2.1.

Unfortunately, this criterion is very difficult to manage, as shown in figure 8.4. The results are non-predictable, and this is probably because the time spent in the front for a voxel is related to the local cost of the propagation at this voxel, but do not have any relation with the position of the voxel relatively to the object that we

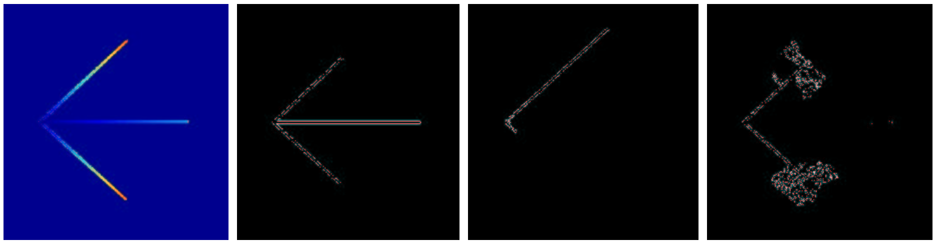


Figure 8.4. Instability of the Time criterion for Freezing: Left image is the action map obtained with a maximum time criterion of 100 iterations; Other images are freezing maps (white pixels) with respectively from left to right 100, 80 and 60 iterations as maximum time spent in the front.

are trying to segment.

Using Distance for Freezing

The distance to the start point is a direct output from the method we already developed for reducing user-intervention in the *Virtual Endoscopy* process in sections 2.2.3 and 3.1. It seems far more "natural" to use the distance to the starting point, or relatively to the most far propagating part of the front, since this notion is completely embedded in the topology of the object we are trying to extract: the section of a tube-shaped objects must be small towards its extent. We must discriminate the points of the front that are near the initializing seed points while other parts of the front are already far. It will prevent from flooding in non-desired area of the data.

We can fix several criterion for the *Freezing* based on the distance. Knowing the current maximum Euclidean path length d_{max} in the front propagation process we can decide that a voxel $\mathbf{v}$ of the propagating front (i.e. *Trial* voxels) should be removed from the front (i.e. set as *Alive* voxel):

- if $\mathcal{D}(\mathbf{v}) < d_{max}/\alpha$, with $\alpha \geq 1$ user-defined; or
- if $\mathcal{D}(\mathbf{v}) < \max(d_{max} - \tilde{d}, 0)$, with $\tilde{d} > 0$ chosen.

The results are now predictable, in the sense that the Euclidean distance to the starting point is a measure which contains information about the geometry of the surface extracted, and in particular its length. This is less related to the local cost of the propagation in each voxel, and more to the position of this voxel in the object. This distance criterion has proven reliability as well in 2D as in 3D, and we worked upon its implementation in the following. A 2D example on the synthetic test is shown in figure 8.5.

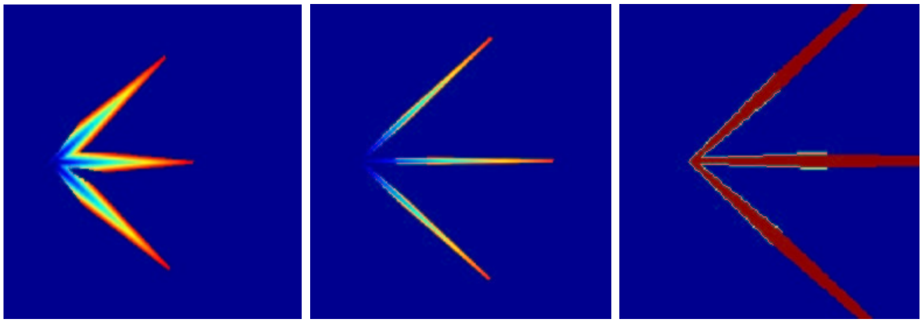


Figure 8.5. Using Distance for Freezing: Left and middle images are action maps with distance criterion of respectively 100 and 50; right image is a zoom on the freezing map for a distance criterion of 50: the pruned points are set in green.

Algorithmic implementation of the *Freezing*

Once the criterion has been chosen, at each time step we insert our visited points both in the classical action related heap, and in another data-structure where the ordering key is the criterion. As for the action, we can use a min-heap data-structure, since the partial ordering provided is sufficient.

At each iteration, we are able to remove all the points whose keys are greater/lower than this criterion, starting from the minimum/maximum element in the tree. It can be implemented easily for the time criterion by recording the iteration at which any point has been inserted in the heap, and to store this time in another min-heap data-structure. Therefore, the element at the top of the heap will still be the point that has spent the longest time without being evolved to the *Alive* set. For the distance criterion, the min-heap key is the computed distance, which means that the element at the top of the heap will still be the point that is the nearer *Trial* point to the starting point.

In the following is detailed an algorithmic implementation of the *Freezing* with the second criterion for the distance information.

Definition

- a starting point $\mathbf{p}_0$, located at the root of the tree structure;
- the usual set of data-structures for front propagation, including an action map $\mathcal{A}$, one min-heap structure $\mathcal{H}_{\mathcal{A}}$ and a penalty image $\mathcal{P}$ which will drive the front propagation, and which is a function of the position only;
- a distance map $\mathcal{D}$ to compute the Euclidean minimal path length, as explained in section 2.2.3;
- another min-heap data structure $\mathcal{H}_{\mathcal{D}}$, where the ordering key for any point $\mathbf{p}$ is the value of $\mathcal{D}(\mathbf{p})$, which means that the first element of this heap will be the *Trial* point with smallest distance $\mathcal{D}$;
- several counters d_{max}, $\tilde{d}$, d_{stop}

Initialization

- initialize the classical front propagation method, setting $\mathcal{A}(\mathbf{p}_0) = \mathcal{D}(\mathbf{p}_0) = 0$ and storing the seed point $\mathbf{p}_0$ in both min-heap structures $\mathcal{H}_{\mathcal{A}}$ and $\mathcal{H}_{\mathcal{D}}$;
- $d_{max} = 0$
- $\tilde{d}$ and d_{stop} are parameters for tuning the algorithm (user defined).

Loop: at any iteration

- Let $\mathbf{p}_{min}$ be the *Trial* point with the smallest action $\mathcal{A}$;
- proceed according to the classical *Fast-Marching* algorithm, by examining its neighbors, and updating the min-heap $\mathcal{H}_{\mathcal{A}}$ with the new action values computed;
- take $d_{max} = \max\left(d_{max}, \mathcal{D}(\mathbf{p}_{min})\right)$;
- consider $\mathbf{q}_{min}$, the first element of $\mathcal{H}_{\mathcal{D}}$, being the *Trial* point with the smallest distance $\mathcal{D}$. While $\mathcal{D}(\mathbf{q}_{min}) < \max\left(d_{max} - \tilde{d}, 0\right)$ do
    - set $\mathcal{D}(\mathbf{q}_{min}) = \mathcal{A}(\mathbf{q}_{min}) = \infty$;
    - set $\mathbf{q}_{min}$ in the *Alive* set, then $\mathbf{q}_{min}$ will not be used for computing the action/distance at its neighbors location.

  - delete $\mathbf{q}_{min}$ in both $\mathcal{H}_{\mathcal{D}}$ and $\mathcal{H}_{\mathcal{A}}$;
- if $d_{max} > d_{stop}$, exit the loop.

This heuristic is to discriminate the parts of the front that are propagating slowly, by recording the maximum distance which has been traveled, and compare it to the distance which has been traveled by this parts. If the ratio between those two distances is two important ($>$ given threshold), we "freeze" those parts by setting there speed artificially to zero. It enables to stay inside the object when it is long and thin like tubular structure, as shown in figure 8.5. The domain visited by our algorithm is slightly smaller than the previous one (figure 8.4-right) and this domain shortens with the distance criterion, when we compare left and middle images in figure 8.5. The figure 8.5-right clearly demonstrates than the *Freezing* principle discriminates the points located far from the propagating fronts.

Illustration on the Vascular tree extraction problem

The method explained previously is very useful when it is used for vascular segmentation. Initialization step is therefore performed in a very fast manner by just setting a seed point at the top of the tree hierarchy. Figure 8.6 displays results of this algorithm. The distance threshold is a parameter which is not very sensitive: we generally take a value related to the *a priori* dimensions of the object. This threshold must be more important than the assumed maximum section of the object. It will approximately represent the volume of points bounded by connected envelope of the front voxels that are not frozen.

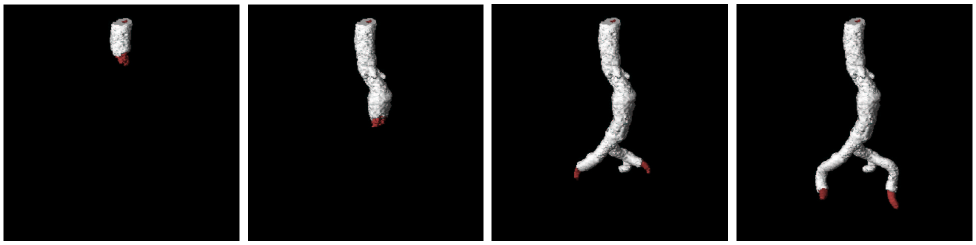


Figure 8.6. Using Distance for Freezing in the Aorta: From left-to-right, images show iterations of the segmentation process; the propagating front is in red, and the frozen voxels are in white.

8.3.2 Suitable stopping criterion

Having designed an adequate criterion for *Freezing* the unwanted parts of the front that could lead to “flooding” of the evolving wave in other parts of the image, we now explain our strategy to stop automatically the propagation.

Previous strategy was to use a maximum Euclidean path length to stop propagation, like for the virtual endoscopy application. In *Virtual Endoscopy* , the user can set both extremities of the trajectory, if he has an *a priori* knowledge of the anatomical objects. Extraction of tree-like structures cannot use such an assumption: the number of branches in our structure is undefined, only assumption being that the user can fix a point inside the structure, at the beginning of the segmentation process.

The *Freezing* process will provide a criterion which is independent of the number of different branches to recover. If we plot the maximum distance d_{max} of section 8.3.1, as a function of iterations while propagating, we will observe the following profile shown in figure 8.7. We clearly see that this distance increases linearly until a big decrease of the slope appears. It is important to notice that this shock indicates when the front flows out of the object at "heads" of the front. We decide to stop front propagation at this particular time. During the first part of the plot, the function is quasi-linear. The slope is directly related to the section area of the tubular object. By definition of Fast Marching, the number of iterations is equal to the number of voxels that are alive. It means that passing through a certain length in the aorta

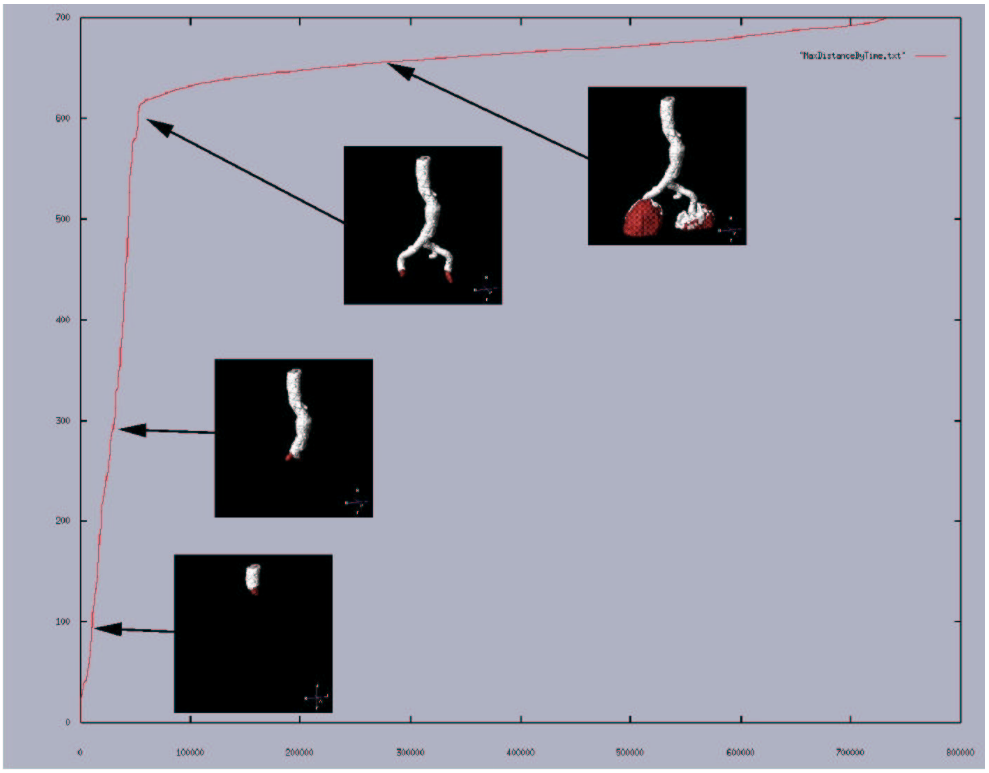


Figure 8.7. Using Distance for Stopping propagation in the Aorta: The images of the propagating fronts of figure 8.6 are super-imposed on the evolution of the maximum distance crossed by the front propagation across iterations; it emphasizes that the decrease in the slope is related to the "flooding" out the aorta.

implies to visit a number of voxels proportional to the length.

Let us assume that the global section of our aorta is constant in our dataset. This is approximately true in large parts, but becomes a wrong assumption in the very thin parts of the vessels and arteries. But we can assume that the front propagates at the same speed inside the object. Therefore, the number of voxels visited is proportional to the section area. Then the slope collapse can be easily detected using a simple threshold on the slope, depending on the object we want to extract. Even if there are aneurysms in the data set, and even if the mean section of the object increases with the depth, we can assume that we do not want to extract an object which is twice the maximum section. We could then derive a criterion on the maximum section of the object S_{max} which is obviously related to the section area of the object of interest. Recording the first iteration where the front flows out, it gives us the maximum distance where we must stop propagation.

8.4 Extracting the skeletal information

In the following, we assume that we use the *Fast-Marching* and the *Level-Sets* in a collaborative manner, in five steps:

1. the user input is a seed point for region-growing;
2. the *Fast-Marching* using the *Freezing* principle is evolved from this starting point;
3. this evolution is stopped using either the distance, the user intervention, or an automatic criterion;
4. the binary mask defined by the propagation gives the initialization of the region based descriptors k_{in} and k_{out}, as used in section 6.2;
5. the *Level-Sets* model is evolved with equation 4.6 for a small number of iterations.

The process is really similar to the framework detailed in section 5.4. The *Fast-Marching* using the *Freezing* principle will act as a rough initialization step, which will provide the binary image of the voxels visited. This mask will also serve to initialize the different probabilities of the region descriptors defined in section 5.2. First row in figure 8.8 shows the surfaces of several tubular structures extracted with the *Freezing* algorithm. The domain of voxels visited during this first step is used to set correctly the descriptors of the *Level-Sets* model, that converges in a few iterations to the surfaces which are shown in the second row in figure 8.8. Notice that the scheme used here is in equation (7.1), where forces have been included to restore the distance function.

In this chapter, we do not implement dedicated algorithms based on the *Level-Sets* methods. They are used in a very classical manner, to converge to sub-pixel accuracy results, on the basis of as-hoc fast algorithms. However, the level of accuracy that is achieved by the *Level-Sets* cannot be of course outperformed by the initialization

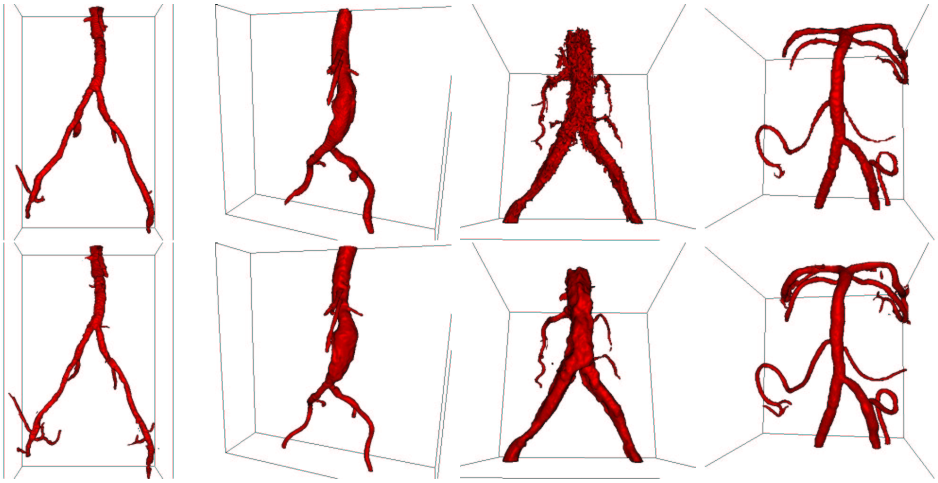


Figure 8.8. Final segmentation of vascular objects: First row shows different vascular objects that have been extracted with the *Freezing* algorithm - except the example shown in last column of the right, where the method used was the competitive fronts algorithm; Second rows is the final result of the segmentation after 40 iterations.

method. The convergence step they achieved cannot be replaced in any way by the *Fast-Marching*.

8.4.1 Combining path and shape extraction

The complete framework for path and surface extraction we have developed will be illustrated in this section by results on a **3D-RA** acquisition of a stenosed vessel, which is shown in figure 8.9.

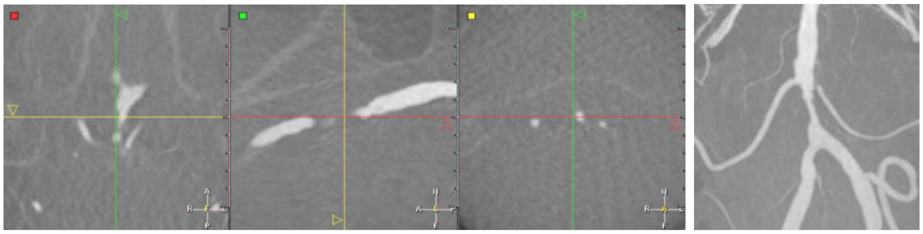


Figure 8.9. 3D-RA dataset of an aortic stenosis: left image shows three orthogonal views of the dataset; right image is a **MIP** view of the same dataset.

We have shown in the part I of the thesis how to extract a trajectory inside a tubular structure. In part II of the thesis, we have combined fast and accurate methods for shape extraction of tube like objects. We now want to combine the results of both parts, and extend this facility to the detection of tree-like structures, and their corresponding set of multiple trajectories, in order to enhance measures and visualization of pathologies of any tube-shaped object.

We worked upon the extension of the trajectory extraction method, applied for example to virtual endoscopy in chapter 3 to the case of multiple trajectories. For example, the dataset in figure 8.9-left is a structure with branches, which pathology - a stenosis - is clearly visible on the **MIP** view in figure 8.9-right. The complete study of this pathology, with minimum interactivity, would be to extract its surface, and all needed trajectories inside it, in order to give accurate measurements.

Techniques found in the literature

The combination of path and shape representation is a framework already studied as well in Computer Graphics as in Computer Vision. In Computer Graphics, cylindrical shapes description is done by implicit surfaces (in the sense of [13, page 223]) defined by the convolution of a filter kernel with a skeleton. In other words, this *distance surface* is a surface that is defined by distance to some set of skeletal elements, like any curve. But in graphics, the target is to improve visualization and interactivity over the representation of the object. However, it connects to vision because it is often convenient to model a shape as a generalized cylinder as done in [132], for reconstruction of anatomical shapes, as done in [175] by combining the fitting of a generalized cylinder, and its symmetry axis.

In those methods, the central axis constrain the extraction, and models the tubeness of the final object extracted.

Our multiple path extraction method

In our case, the shape is initialized by *Fast-Marching*, thus a path construction method, but we are going to use the solution at convergence of the *Level-Sets* in order compute the final set of trajectories - i.e. the skeletal information of our object. Therefore, shapes controls path extraction. This is exactly the kind of methods that lead to accurate measurements and visualization of the objects:

1. It relies on a sub-pixel shape extraction model; thus the intersection of a cross section plan and the surface is an improved measure of the objects, while cylinders approximate the model.
2. The *Level-Sets* enables any change in topology, and there is no constrain on the initialization of the model, how huge can be the number of branches in the anatomical object.
3. The paths used for quantification are based on this robust surface extraction model, increasing the robustness of the measures.
4. The user input is limited to the setting of the root of the tree hierarchy.

Our method is based on the construction of a connectivity map, by looking at several chosen iterations to the connectivity of the propagating front (i.e. the *Alive* voxels)

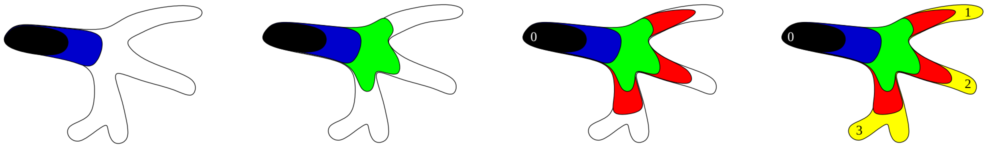


Figure 8.10. Multiple path extraction algorithm: From left to right, these are connectivity tests made on the propagation of a wave inside a segmented object, starting from the voxel designated by label 0, until the whole domain is visited.

and the connectivity of the sets of voxels visited, as shown in figure 8.10. Defining a distance step, each time this step has been accumulated by the front, we label the different sets of visited voxels, and we thus detect when a front separate, at a branch, into several not connected sets.

When the whole domain has been visited, we take for each separate set a representative voxels, which is the most far from the starting point, and we set it as an applicant extremity for back-tracking a trajectory. Notice that the distance step defines the accuracy of the method, since a too important step will lead to misunderstandings: on the right image, only the extremity designated by the label 3 will be eligible for back-tracking, while there are two branches, because the distance step is bigger than both branches.

Multiple Path Extraction Algorithm

The algorithm we devised for multiple path extraction is mostly inspired from works on skeletal extraction from binary, or scattered data. It can be easily compared to morphological processes, but has two advantages: we can choose the *scale* or accuracy of the multiple path extraction, and we can derive this *scale* from anatomical knowledge of the data studied. It is a complete framework in the sense that, the path extraction relies here on a segmentation process which can be as well handled by the *Fast-Marching* or the *Level-Sets* methods. This segmentation step defines a binary mask $\mathcal{M}$ which is one of the main input of our algorithm:

Definition

- a binary mask $\mathcal{M}$ which defines the region of interest in the image;
- a penalty image $\mathcal{P}$ which will drive the front propagation;
- a distance map $\mathcal{D}$, computed with the method described in section 2.2.3, and a distance step d, user-defined parameter that controls the accuracy of the end-point extraction;
- a counter c_d that recalls the iteration number of the loop in our algorithm;
- a label map $\mathcal{L}$ to label each branch detected, $n_{\mathcal{L}}$ a label counter, and an array $\mathcal{E}$ which will recall the hierarchy of the branches detected;
- a starting point p_0, located at the root of the tree hierarchy.

Initialization

- $\mathcal{M}(i,j,k) = 1$ for all voxel in the region of interest, elsewhere $\mathcal{M}(i,j,k) = 0$;

- $\mathcal{L}(i,j,k) = -1$ for all voxel in the image domain, $n_{\mathcal{L}} = 0$, and all elements of $\mathcal{E}[i]$ are set to -1;
- We initialize the usual set of data-structures for front propagation, including an action map $\mathcal{A}$, the distance map $\mathcal{D}$, and a min-heap structure;
- we initialize a classical front propagation method, setting $\mathcal{A}(p_0) = 0$ and storing p_0 in the min-heap structure; item the counter $c_d = d$.

Loop

- we propagate the front with Eikonal equation, computed with penalty $\mathcal{P}$ on the domain defined by the mask $\mathcal{M}$;
- for each *Trial* point p visited in the *Fast-Marching* algorithm, $\mathcal{L}(p)$ is set to the label of its current *Alive* neighbor with minimal action;
- if we visit a voxel p with $\mathcal{D}(p) \geq c_d$:
    1. we consider the set of *Trial* points $\mathcal{T}$, that are all stored in the min-heap data structure, we consider $t_1, \ldots, t_k$ its k subsets of connected components (with 26-connexity in 3D), obtained through a simple connectivity algorithm;
    2. In all subset $t_i, i \in [1, \ldots, k]$
        - considering the old label $l_{i_{old}}$, and the new label $l_{i_{new}}$, we set $n_{\mathcal{L}} = n_{\mathcal{L}} + 1$, $l_{i_{new}} = n_{\mathcal{L}}$, and $\mathcal{E}[l_{i_{new}}] = l_{i_{old}}$;
        - for all the points $p \in t_i$, we set $\mathcal{L}(p) = l_{i_{new}}$;
    3. $c_d = 2 \times c_d$;
    4. we stop if the whole domain defined by $\mathcal{M}$ is visited.

Termination

- we consider all sets $\mathcal{L}_j$, $j \in [1, \ldots, n_{\mathcal{L}}]$ defined by the label map $\mathcal{L}$ with different labels l_j;
- we select the subset of $\mathcal{L}_k$, $k \in [1, \ldots, n_{\mathcal{L}}]$, which have $\mathcal{E}[l_j] \neq -1$ and $\forall n \in [l_j; n_{\mathcal{L}}]$ $\mathcal{E}[n] \neq l_j$;
- $\forall \mathcal{L}_k$ selected, we find the voxel (i,j,k) with maximum distance $\mathcal{D}(i,j,k)$ and set it as end point for back propagation;
- we back-propagate from all final voxel selected and extract a set of multiple trajectories.

Figure 8.11 shows several label maps $\mathcal{L}$ with $c_d = 10$, 30 and 50. c_d is the minimum size of the branches detected, it is the *scale* of the algorithm accuracy. If this scale is chosen small, lots of branches will be detected, but if the scale is increased, the computation time will decrease as well, because it controls the number of connectivity tests which are performed on the *Trial* voxels, during propagation.

illustration on the vascular tree extraction

In figure 8.12, one can observe the complete framework of *Fast-Marching* initialization followed by several iterations, using *Level-Sets* methods, and finally, the extraction of multiple trajectories inside two different datasets. The computations for the paths are restricted to a small number of points, located inside the objects of interest (usually less than 20% of the whole volume, leading to interactive computing times. Those

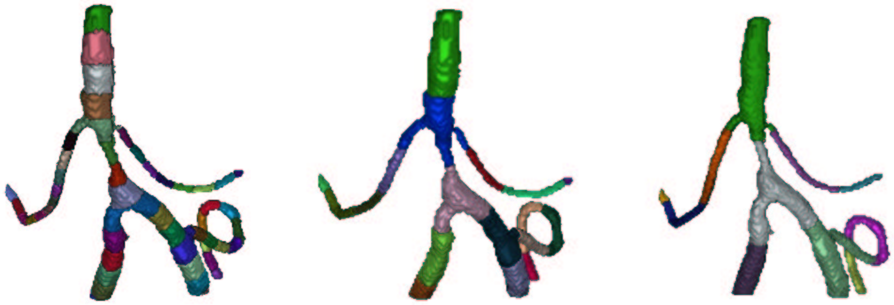


Figure 8.11. Labeling algorithm for multiple path extraction: From left to right, the images show the label map obtained with the multiple path extraction routine applied with path steps 10, 30 and 50 respectively.

paths are already very useful for virtual inspection of pathologies, for example in the aorta (as done in the section 3.1), or measurements along the trajectories extracted, using the techniques detailed in section 7.2. Figure 8.13 shows the result of applying the multiple path extraction algorithm explained previously. This set of paths is the basis of the quantification techniques that can be applied on such a dataset (this aorta presents an Abdominal Aortic Aneurysm).

Originality of this algorithm, towards front propagation techniques applied for multiple path extraction, as in [101], where the set of endpoints is manually drawn in the original image. In our case, all trajectories are extracted automatically.

8.4.2 From Trajectories to Tree Extraction

The trajectories obtained with our algorithm can guide virtual endoscopes. They can also be used for quantification of pathologies, by measuring the variation of the section of the object, across the curvilinear abscissae of the path extracted. But the information of trajectory is not related to the whole branching structure and is just the minimal centered path between two extremities. Therefore, the user is assumed to know the position in the object of this trajectory. And those trajectories are not related to each other, leading to possible misunderstanding in this position. Moreover, this absence of spatial relationship between the paths and the surface disable the use of further developments like automatic labeling of the branches, and accurate localization of pathologies. In order to extract the information which is relevant in order to analyze the surface of the tree-shaped object extracted, we need to extract the underlying skeleton on the basis of our trajectories, as done in figure 8.14. The process of extracting the tree structure from the trajectories is simple: during backtracking of the trajectories, we adjoin those which are close to each other, creating a branching point. The only parameter is the definition of proximity between trajectories.

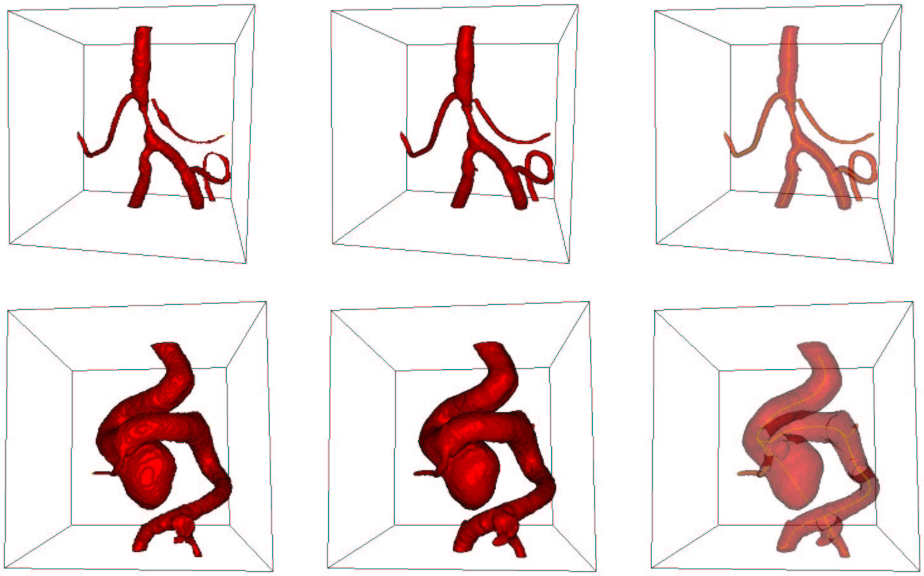


Figure 8.12. Complete method applied to several objects: First row is the framework applied to the stenosed object of figure 8.9 and second row concerns the aneurysm shown in figure 6.4 - Left column is the initialization given by the *Fast-Marching* method; middle column is the surface obtained after a small number of iterations of the *Level-Sets* method; right column shows the multiple trajectories extracted with the labeling algorithm, by transparency.

Algorithmic implementation

As a second process, we can extract a skeleton of our object, from this set of multiple trajectories. The initialization use the same input than the multiple path extraction process, including the final end points extracted.

Definition

- a binary mask $\mathcal{M}$ which defines the region of interest in the image;
- a penalty image $\mathcal{P}$ which will drive the front propagation, usually this penalty map is computed using the centering method described in section 2.3;
- the action map $\mathcal{U}$ computed with this penalty during the initial multiple path extraction;
- the starting point p_0, located at the root of the tree hierarchy;
- the set of end points e_i $i \in [1; N_e]$ where N_e is the number of end points extracted.
- a distance step d which defines the minimum distance between two trajectories (this distance step is chosen bigger than the gradient descent step).

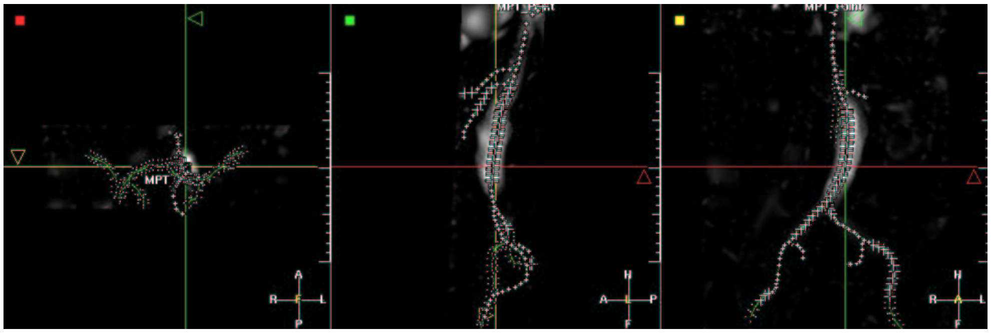


Figure 8.13. Multiple trajectory extraction from only one seed point: This figure represents the projection on three orthogonal views of the complete set of trajectories tracked in the aorta MR dataset which was segmented in figure 8.8 in the second row; the *Freezing* method for initialization with the *Fast-Marching* algorithm has been used to extract centered trajectories.

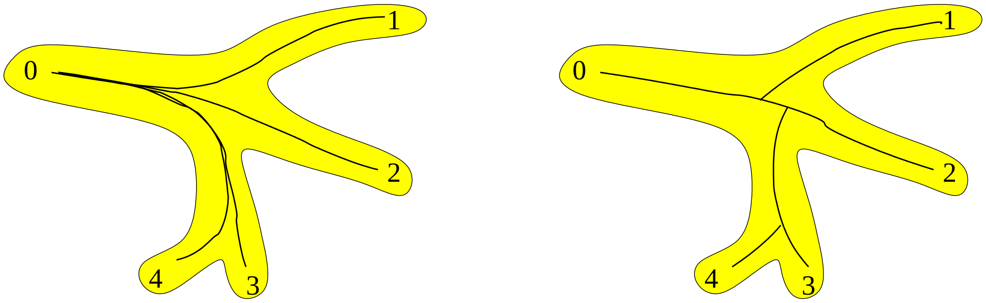


Figure 8.14. From trajectories to tree representation: Left image is a set of trajectories extracted in a segmented object. right image represents the valuable tree structure needed for quantification.

- another different label map $\mathcal{L}$ to label the voxels that are neighbors of a path, which means that the distance between this voxel (i, j, k) and a path extracted is less than d;
- an array $\mathcal{E}$ to recall the branches detected.

Initialization

- $\mathcal{L}(i, j, k) = -1$ for all voxel in the image domain;
- $n_e = N_e$ and $\forall i \in [1; n_e]$, $\mathcal{E}[i] = 0$.

Loop: for $i \in [1; N_e]$

- we back-propagate from e_i, on the action map U using a simple gradient descent method, as described in equation 2.6;
- at every path step, the position of the new path point is defined by $(x, y, z) \in \mathbb{R}^3$
- we consider the vertices of the Cartesian grid that surround $\overrightarrow{x} = (x, y, z)$, the voxels $\overrightarrow{n} = (i, j, k) \in \mathbb{N}^3$ which verify $D_2(\overrightarrow{x}, \overrightarrow{n}) < d$, where D is the Euclidean distance;

- if, for all those vertices $(i, j, k) \in \mathbb{N}^3$, $\mathcal{L}(i, j, k) = -1$, we set $\mathcal{L}(i, j, k) = i$, and continue back-tracking for e_i;
- else, if one of the vertices (i, j, k) verifies $\mathcal{L}(i, j, k) \neq -1$, a branching point is detected, then:
  - recall the label $l = \mathcal{L}(i, j, k)$;
  - $n_e = n_e + 1$, $e_{n_e} = (i, j, k)$;
  - $\mathcal{E}[i] = e_{n_e}$ and $\mathcal{E}[n_e] = 0$;
  - stop back-tracking for e_i;
  - continue back-tracking, this time for e_{n_e}, substituting all $\mathcal{L}(i, j, k) = l$ by $\mathcal{L}(i, j, k) = n_e$, until another branching point or p_0 are found;
- if we reach p_0, then stop back-tracking for e_i.

Termination

- for all end point e_j $j \in [1; n_e]$, we can consider the couples of endpoints $(e_j, e_{\mathcal{E}[j]})$ as extremities of linear parts of the skeleton (with $e_0 = p_0$).
- the multiple paths between couples of points $(e_j, e_{\mathcal{E}[j]})$ $j \in [1; n_e]$ build the skeleton of our object, at scale c_d and distance d.

Figure 8.15 displays the result obtained on the dataset shown in figure 8.9. From the set of multiple trajectories, branching points are extracted, as shown in figure 8.15-right.

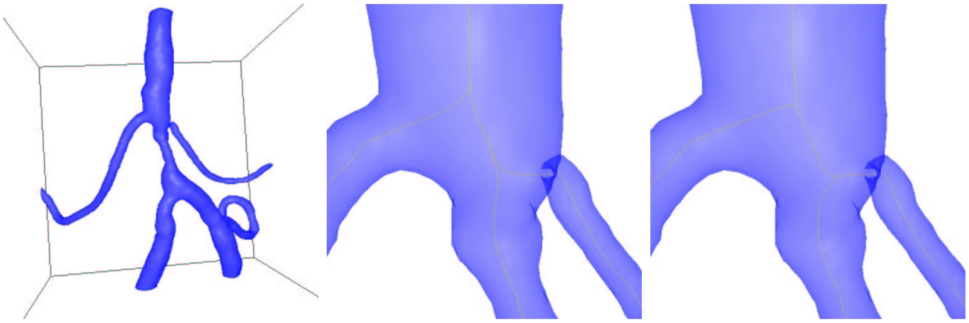
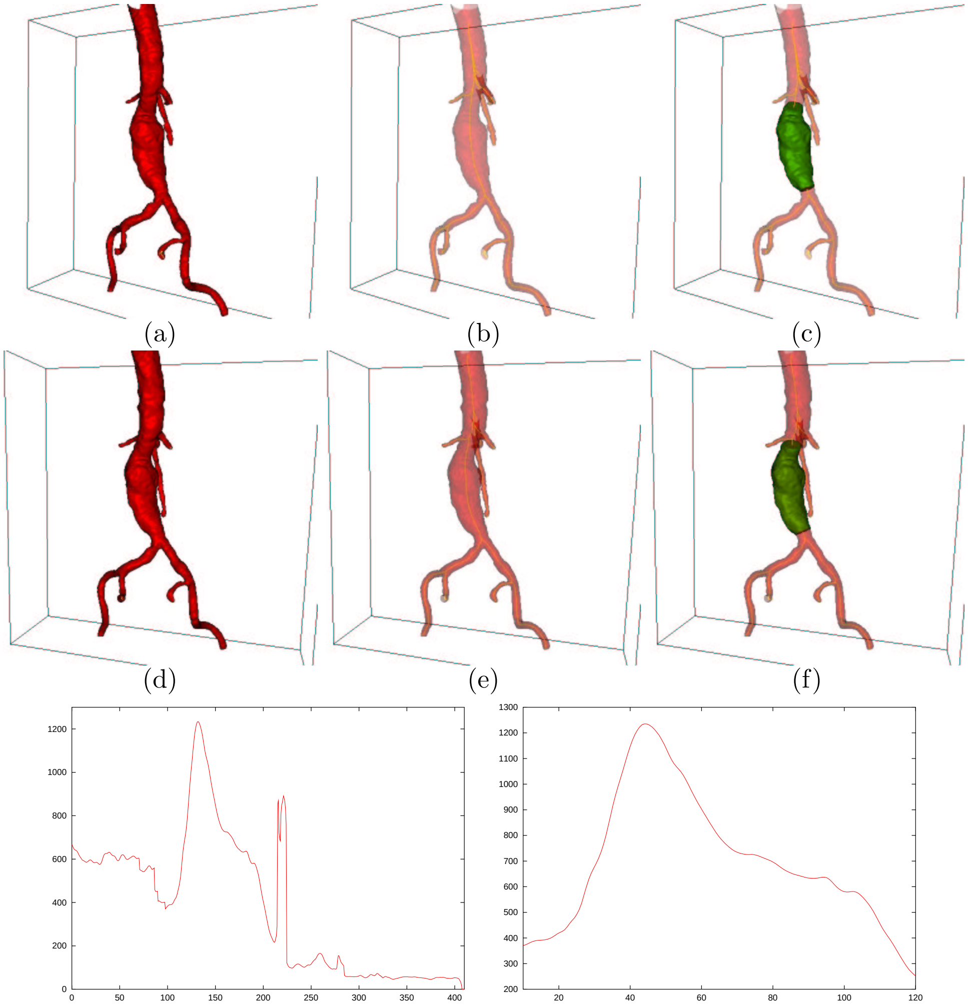


Figure 8.15. Obtaining a tree hierarchy from a set of trajectory: Left image is the segmented object extracted from the dataset shown in figure 8.9; middle image is a zoom on two bifurcations of the object, where the trajectories extracted are displayed; right image is the same point of view on the translucent surface extracted with the tree extracted from the set of paths.

Measurements on the tree

The computational cost of the tree extraction finds its justification in the improvement of the measurements along the new set of trajectories available. Figure 8.16 compares the section measurements with multiple path extraction technique, and tree



(a) (b) (c)

(d) (e) (f)

(g) (h)

Figure 8.16. Comparing results with the multiple path extraction, and the tree extraction: First row shows images of the segmentation process (a) plus the multiple paths extraction, visible by transparency (b) and the region of interest in green that isolate the aneurysm (c) along one of the trajectories; second row shows the same images (d,e,f) using the tree structure extracted from the same seed point; last rows shows the variation of the section along the paths that are inside the aneurysms, for the complete trajectory (g) and for the branch (h).

extraction technique (dataset shown in figure 3.12). The tree extraction, as shown by transparency on figure 8.16-(e) enables to measure the section along the necessary subset of the object, delimited by the the two branching point (this subset has been colored in green on the figure). If this information is plotted across a trajectory in the entire object, it is not useful for two reasons

- section information is not valuable at the branching points;
- the position of the part of interest cannot be obtained straightforwardly.

This problem is illustrated in the last row of figure 8.16. The plot of the object section across the curvilinear abscissae of a trajectory is shown in figure 8.16-(g), versus the same plot across a branch of the tree extracted in figure 8.16-(h).

Chapter 9

Application à l'Extraction, la Visualisation, et la Quantification d'Objets Anatomiques Arborescents

Résumé — A partir des outils de visualisation et de mesure du chapitre 7, et des outils d'extraction de surfaces et de sequelettes du chapitre 8, nous étudions plusieurs cas pratiques, dédiés à des objets particuliers. Dans la section 9.2, nous appliquons notre méthode d'extraction de trajectoires et d'arbre du chapitre précédent à l'extraction et la quantification des bronches dans les images médicales 3D de scanners CT. Dans la section 9.3, on considère un tout autre problème: celui du groupement perceptuel, ou la donnée est un ensemble non-structuré de régions de l'image. Nous proposons une méthode de reconstruction de structures arborescentes dans des images médicales tridimensionnelles.

Abstract — Using the several techniques developed in chapters 7 and 8, we develop applications for medical problems. In section 9.2, we apply the complete multiple paths and shape extraction framework of chapter 8 to the segmentation and quantification of airways in 3D multi-slice CT scanner images. Finally, in section 9.3 we consider the problem of *Perceptual Grouping* and contour completion, where the data is an unstructured set of regions in the image. We propose a new method which is illustrated on reconstruction of tree structures in 3D angiography images.

9.1 Application to 3D Vascular Images with Multiscale Vessel Enhancement

In this section, we focus on vascular tree extraction, for accurate determination of vessel width (important in grading vascular pathologies, such as stenosis, or aneurysm). We are particularly interested in using Multiscale Vessel Enhancement techniques of *Frangi et al.* [60].

9.1.1 Medical relevance

All methods developed in this chapter are illustrated on the particular problem of vascular tree extraction in 3D contrast enhanced medical images. The medical interest of this extraction is mostly accurate determination of vessel width. It is an important step in grading vascular pathologies, such as stenosis, or aneurysm.

Stenosis quantification

In the carotid arteries, this quantification determines the choice of stroke treatment. Studies have revealed that patients with severe symptomatic stenosis in the carotids should undergo surgical treatment, and support the relevance of accurate measurement techniques of vascular segments.

Aneurysm quantification

For explanations on this pathology we refer to section 6.1 where they are studied in the case of cerebral vessels. Those pathologies, which are roughly speaking “inflations” of an artery that weak its walls, and can lead to an hemorrhage, occur for example in the brain and, and in the abdominal aorta (see figure 3.12).

Potential: Multiscale Vessel enhancement

For the definition of the speed function for the *Fast-Marching* algorithm, we can use the output of a multi-scale vessel filters based on the Hessian matrix [105, 60]. This paragraph will be illustrated by an application on the dataset shown in figure 9.1[1]. We have used the measure defined by *Frangi et al.* [58] in the following. The symmetric Hessian matrix $\mathbf{H}$ describes local second order intensity variations in the image and is given as:

$$H_{ij} = \frac{\partial^2 I}{\partial x_i \partial x_j} \ , i,j = 1,\ldots,n \tag{9.1}$$

where $I(\mathbf{x})$ is the n-dimensional image. The Hessian matrix defines an ellipsoid where the direction of its smallest axis is the direction of minimal second derivative, that defines the local direction of a tub-like structure. Having extracted the three eigenvalues of the Hessian matrix computed at scale σ, ordered $|\lambda_1| \leq |\lambda_2| \leq |\lambda_3|$, we define

---

[1] We would like to acknowledge Dr Wiro Niessen, from Image Sciences Institute, University Hospital Utrecht, Netherlands, who provided this image.

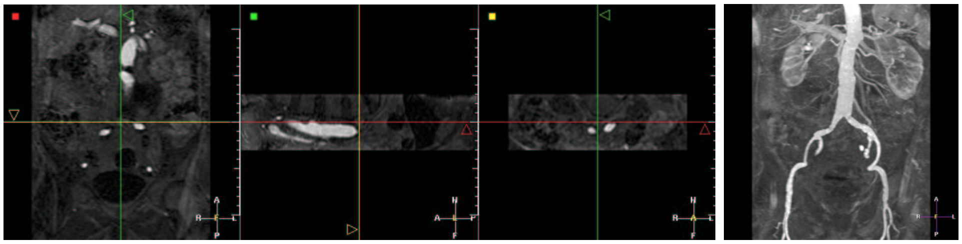


Figure 9.1. Contrast (Gd-DTPA) MRA image of the aorta: Left image shows three orthogonal views of the dataset; right image is a **MIP** view.

a vesselness function

$$\nu(s) = \begin{cases} 0, \text{ if } \lambda_2 \geq 0 \text{ or } \lambda_3 \geq 0 \\ (1 - \exp\frac{-{R_A}^2}{2\alpha^2}) \exp\frac{-{R_B}^2}{2\beta^2} (1 - \exp\frac{-S^2}{2c^2}) \text{ else} \end{cases}$$

where the ratios $R_A = \frac{|\lambda_2|}{|\lambda_3|}$ and $R_B = \frac{|\lambda_1|}{\sqrt{|\lambda_2 \lambda_3|}}$ are used to distinguish between lines and sheet-like structures and to measure deviation from blob-like structures. These measures arise from geometric interpretation as

$$\begin{aligned} R_A &= \frac{|\lambda_2|}{|\lambda_3|} = \frac{\pi|\lambda_2\lambda_3|}{\pi\lambda_3^2} = \frac{\text{largest cross-sectional area}/\pi}{(\text{largest axis length}/2)^2} \\ R_B &= \frac{|\lambda_1|}{\sqrt{|\lambda_2\lambda_3|}} = \frac{(4\pi/3)|\lambda_1\lambda_2\lambda_3|}{(4\pi/3)((1/\pi).\pi|\lambda_2\lambda_3|)^{3/2}} = \frac{\text{Volume}/(4\pi/3)}{\text{largest cross-sectional area}/\pi^{3/2}} \end{aligned}$$

and $S = \sqrt{{\lambda_1}^2 + {\lambda_2}^2 + {\lambda_3}^2}$ is used to reduce influence of the noise due to intensity variations in the background. See [60] for a detailed explanation of the settings of each parameter in this measure.

In figure 9.2 you can observe the response of the filter, based on the Hessian information, at three different scales: $\sigma = 1, 2, 5$. Using this information computed

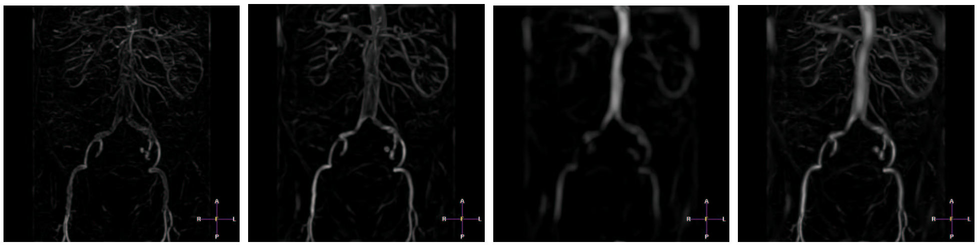


Figure 9.2. Ridge detection in the aorta image From left to right, the measure obtained at three different scales ($\sigma = 1, 2, 5$) and the maximum of the filter response across all scales (MIP visualization of the 3D images).

at several scales, the multiscale response of the filter is the maximum of the response of the filter across all scales, which is shown in figure 9.2-right.

Having computed this measure, we simply use it as speed function, thus propagating faster in the areas with higher filter response, taking

$$\|\nabla T(\mathbf{x})\| = \frac{1}{\max_{\sigma_{min} \le \sigma \le \sigma_{max}} \nu(\sigma, \mathbf{x}) + \nu_{min}} + w \tag{9.2}$$

where σ_{min} and σ_{max} are the minimum and maximum scales at which relevant structures are expected to be found, ν_{min} is just a constant which ensures that speed remains strictly positive, and w is the usual offset term introduced in [34]. This potential will be adapted to bright vascular structures on black background. This potential gives more information than the simple grey level value, since the filter response is higher in the center of the vessel.

9.1.2 Proposed solution

Initialization: *Freezing* method The use of the *Freezing* improves the resulting segmentation: Figure 9.3 shows the difference of segmentation obtained with (right image) and without pruning (middle image). Figure 9.3 demonstrates that the combination of multi-scale vessel enhancement and freezing enhances the segmentation ability of the *Fast-Marching*.

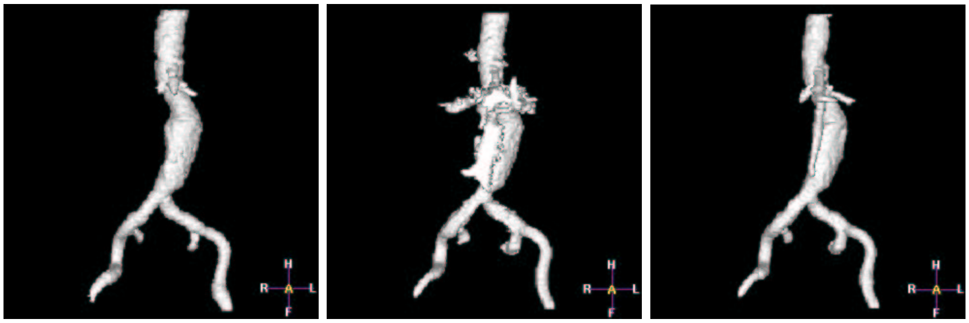


Figure 9.3. Comparing classical and freezing propagation in the Aorta: left image shows the resulting volume obtained using the *Fast-Marching* with a penalty model $\mathcal{P}(\mathbf{x}) = \max(I_{mean} - I(\mathbf{x}), 0)$ where I_{mean} is the mean value inside the aorta; middle image shows the result of a wave propagating in the Aorta MR dataset with a speed based on the Hessian eigenvalues; right image shows the same result using the *Freezing* approach of section 8.3.

Stopping: *Freezing* method This section is illustrated with figure 9.4[2]. Once the multiscale information is available, we can recall the same data than in the details on

---

[2] We would like to acknowledge Dr Wiro Niessen, from Image Sciences Institute, University Hospital Utrecht, Netherlands, who provided this image.

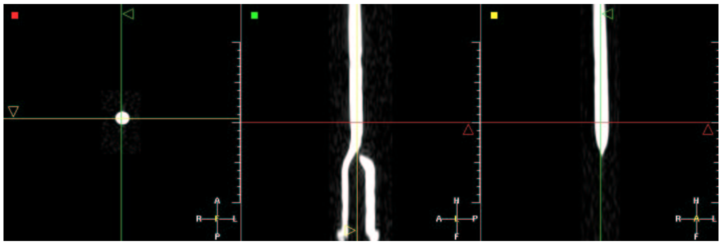


Figure 9.4. Phantom of a stenosed carotid artery in computed tomography angiography (CTA): This example has been chosen for illustration of the stopping criterion, by propagating from the top of the object with a speed $F = 1\ (1 + \|\nabla I\|)$.

the stopping criterion in the previous chapter. If we plot the maximum distance d_{max} of section 8.3.1, across iterations while propagating, we will observe the following profile shown in figure 9.5. A great advantage of this multiscale information is that

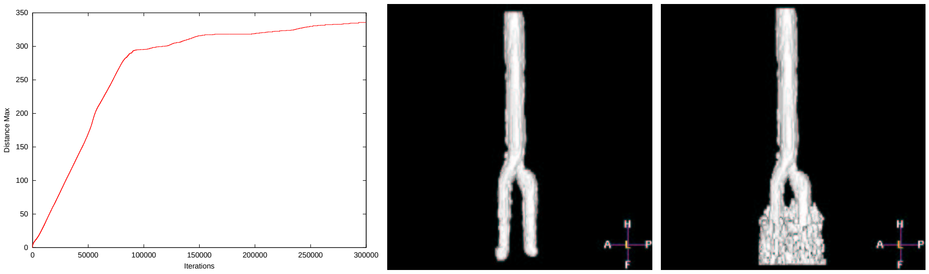


Figure 9.5. Detecting when the front floods outside the object: Left image is the plot of the maximum distance d_{max} across iterations; it is clearly visible that there is is an import change in the slope of the function around a distance maximum of 300; middle image represents the domain of voxels visited when $d_{max} = 290$, and right image is the same for $d_{max} = 310$.

it can be used as a potential for obtaining centered trajectories, with no need to compute the distance to the object walls.

9.1.3 Comparisons and conclusion of the tree extraction method

Other methods for skeletal representation

In [179], authors build a skeletal representation of an unorganized collection of scattered data points lying on a surface. They capture branching shapes, using a distance step similar to ours, by computing the k level-sets from the user-defined root of the tree; and for each of those level-sets, they extract the centroids of connected components. In our case it is not necessary to extract the centroids, because it introduces uncertainty in the location of the branching points. With a centering penalty $\mathcal{P}$, we *aggregate* the paths that are under a user-chosen distance d. This method based on the centroid extraction can be compared to the very interesting work of *Angella* found in [5, 6], which present a deformable and expansible tree as a skeleton extractor, where each node of the tree is a free particle that propagate into the data, pushed by repulsive forces coming from other particles and contours. The set of free particles describes the tree hierarchy. In our case, the sub-voxel precision is very important for visualization and measurements (see sections 7.1 and 7.2), and the needed number of particles for achieving this task would lead to huge computing times.

Very similar work can be found in [157], where the author use wavefronts to extract morphological descriptions of binary images, in particular binary tree structures. However, bifurcations are detected on a projected image in 2D, then this information is upgraded to 3D, but still the method is applied iteratively, looking for bifurcations at each iteration. Our method use a scale parameter c_d, as a distance step in our wavefront, only looking for bifurcations every time the front has crossed a multiple number of this distance. It reduces greatly computations, and can be parameterized by the user, who can only look for branches lower than a typical value d_{min} which is the upper-bound of our scale parameter.

Morphological techniques, like those in [157], are the main tool used for tree extraction, and lots of techniques, like thinning algorithms are already used in medical imaging. They start from volume images so that the traditional medial axis transform of *Blum* [15] can be applied, as in [130, 143]. However, the purpose of our application is to have a smooth set of multiple trajectories. This smoothness is needed for accurate measurements and visualization along the trajectories. Morphological techniques require post-processing in to remove undesirable small parts of the skeleton. Smoothing and removing undesirable small parts of the skeleton is done using our distance step and is very similar to techniques shown in [173], where the scale is also an input in the algorithm. To conclude with the use of morphological techniques, the skeletal description we are looking for corresponds to the needed of an accurate basis for observation and measurements of pathologies. We thus need a smooth and accurate information: a tree which describes the cylindrical topology of the object observed. The variation of the section of a tubular shapes leads to error in medial axis transforms, and to the need of post-processing techniques, to *clean* the skeleton obtained, that our method does not need.

Most impressive work on vascular quantification among others can be found in the PhD thesis of *Frangi*. He develop a very interesting method based on path and shape extraction in [58]:

- The author first set the two extremities of a path on the surface obtained through a iso-surface extraction process;

- the minimal path is extracted on the representation of the surface, using a technique similar to [90];

- a centering force, based on multi-scale enhancement filtering (see [60]) drives the minimal path in the center of the tube-shaped object;

- a circular cross section approximating the vessel is swept along the central vessel axis extracted previously (*swept surface*), and creates a deformable cylinder;
- this cylinder initiates a tensor product B-spline surface [142], that fits the boundaries of the vessel.

Using both path and shapes representation in the same framework, Frangi proposes an elegant method for quantification of vessel morphology [59].

In order to highlight important benefits of our method, we are going to compare our results, that are not completely dedicated to quantification of vascular diseases in MRA images. However, our method proposes an alternative that may overcome several drawbacks of his method.

Topology of the objects: In [57] the bifurcations in carotid arteries introduce errors in the measurements of the stenoses; with our method, bifurcations are localized and wrong measures near branching points can be omitted.

Branching points: We provide the measures in the whole set of branches of our objects, setting a unique tree root seed for segmentation and path extraction. In [57] *Frangi* gives the measure between the defined user end points (he gives also an interesting study of the variability of the results across the user initialization in [57]). In our case, only one point is needed. It enables to reconstruct the whole set of trajectories inside the object, but it converts this information into a tree hierarchy, where important information can be separated from the whole.

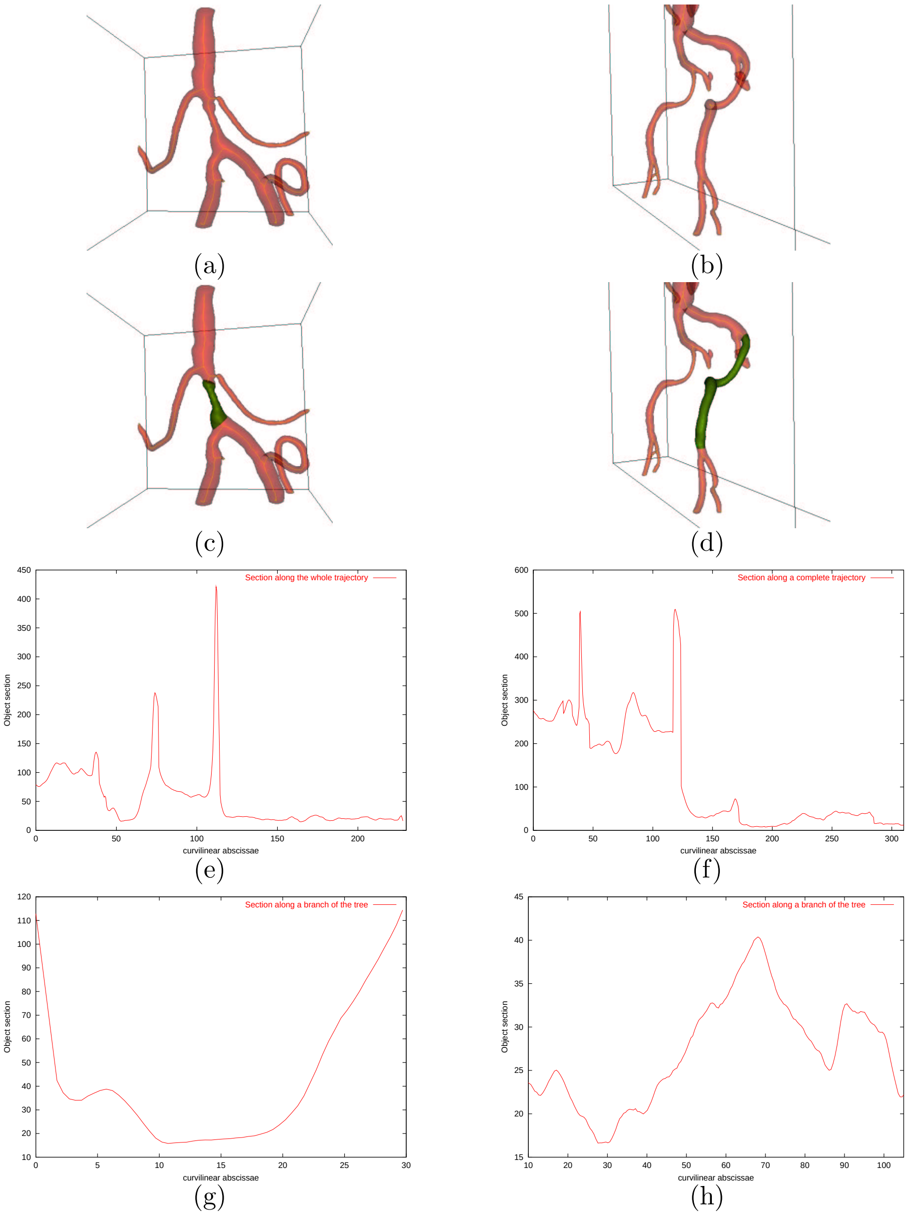
A result of this property of our method is shown in figure 9.6. In particular, figure 9.6-(g) is the information contained in the interval $[40;60]$ in figure 9.6-(e). It is the same process for figure 9.6-(h) which corresponds to the sub-plot contained in the interval $[190;250]$ in figure 9.6-(e). Therefore the tree extraction enables to localize accurately the information needed, as the stenosis extent for the case presented in the left column of figure 9.6.

Accuracy of the model the B-spline that extracts the vessel boundaries in [58] is an approximation of the surface, whereas the zero-level set embedded in $\tilde{\phi}$ has sub-pixel precision.

Conclusion on the vascular extraction method

We have finally a method which provides a sub-pixel information of the position of the shape. Based on the paths extracted with our fast and robust algorithm, the quantification rely on an accurate centered position of the path points. Thus measures and visualization are enhanced (see figure 9.7).

At a matter a fact, this visualization, once paths and shapes are extracted, is real-time, due to the fast rendering of the triangulation of our implicit model. Thus, camera trajectory is managed via the paths extracted. A further extension of this work could be to derive an interface to choose between each branch where to go inside the model.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 9.6. Comparison of measurements on the tree and on a trajectory: Two different datasets are presented, each one in a column (left column dataset present a stenosed vessel). First row (a,b) displays segmented surfaces and extracted trees. Second row (c,d) displays the sub-volume of interest in both cases where sections are performed. Third row (e,f) shows plots of the section measured across the curvilinear abscissae of a trajectory. Fourth row (g,h) displays the same result using branches extracted between two bifurcations.

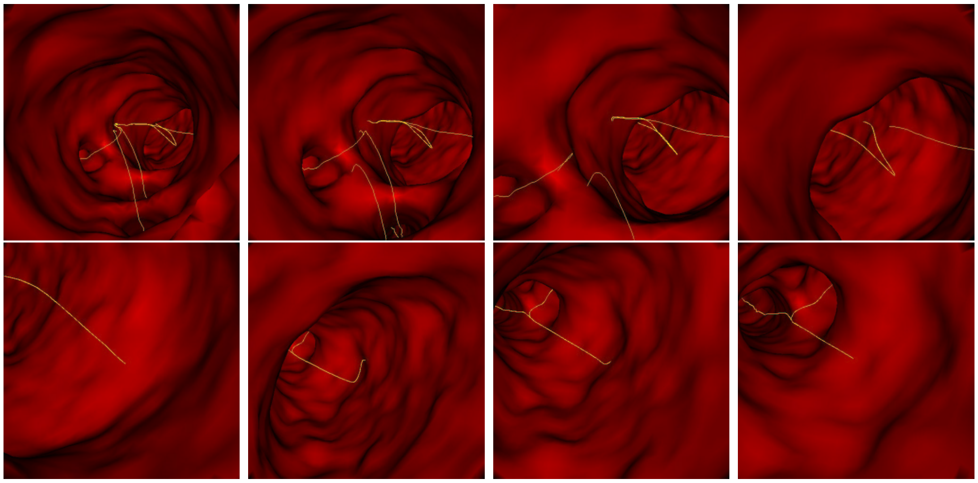


Figure 9.7. Endoscopic view along one trajectory: The whole set of trajectories is displayed (in yellow) simultaneously with the surface rendering: the user do not miss any bifurcation since he can see any branch.

9.2 Application to the Bronchial Tree

9.2.1 Medical interest

Extraction of the bronchi and the bronchial tree

First interest of this segmentation is to provide trajectories for a virtual bronchoscopy system, on the same basis than the virtual colonoscopy tool detailed in section 3.1.

A second important implementation possible is airway tree measurements that can be used to detect lesions or stenoses, structural abnormalities, and to evaluate airway reactivity to external stimuli (for example, evaluating asthma impact on the airways diameters). One must take into account the perspective distortion inherited from the volume rendering, for optimizing stent fitting for example, and the segmentation of the complex bronchi can provide an accurate information.

The framework we developed for the vascular tree extraction can be transposed to this different problem, in order to obtain both surfaces and centerlines of the airways, in typical datasets of the lungs, like multi-slice CT scanner, where voxels are nearly isotropic.

Role of the Virtual Bronchoscopy

The bronchoscopy technique has existed since 1897 and represents probably one of the most frequently used invasive procedures. Even in the hands of a clinically experienced pulmonologist, there is a risk for the patient. However, the goal of virtual bronchoscopy (see an example in [75]) is not to replace real bronchoscopy, which has high advantage of providing a direct inspection of the natural pigmentation which can clearly indicate a pathology. And inability to perform biopsies gives virtual bronchoscopy the role of a detection and much less a characterization technique.

Possible pathologies visible on the multi-slice CT scanner are for example tumors. Malignant tumors of the lung represent the most frequent cause of cancer death in males (35%) and female (18%). Since those tumors (benign or malignant) are only visible on the renderings system when they imply a morphological alteration of the bronchial wall, virtual bronchoscopy cannot contribute substantially to characterization of tumors. Thus, evaluation of virtual bronchoscopy should be restricted to the morphology of the bronchi and direct visualization of intraluminal masses.

However, one handicap of real endoscopy is its inability to see through the bronchial wall, whereas all information surrounding the object is available in virtual bronchoscopy. This handicap is very important since a clinician would like to plan a biopsy in a location that can be accessed only from the bronchial tree. In this case, the virtual bronchoscopy enables to determine in advance the optimal access point for the biopsy procedure.

A further indication for this process is the rare case when the real inspection is contraindicated, as in the presence of a strong stenosis of a branch, or as in the presence of an infiltration due to an extensive tumor manifestation, or in the case of an application in pediatrics, where the necessary sedation can be contraindicated.

Last important improvement brought by the virtual procedure is its clinician teaching device aspect [18, 125]: it can contribute th education and qualification of operating personnel (which benefits the patient by the way).

9.2.2 State of the art in Bronchoscopy imaging

Acquisition techniques

Computed Tomography represents the standard examination technique of the thoracic area, because a natural contrast exists between air and soft tissues, explaining why the trachea and the bronchial tree are perfectly suited for the generation of a virtual bronchoscopy. Three different types of CT data can be used:

1. Incremental CT: a slice is imaged in the axial orientation, after which the patient is shifted to the next position in order to image the next slice. 3D reconstruction (and hereafter renderings) can be calculated from incremental data only when the patient lies so still than no motion occurs during the whole examination. These data are of little use for virtual bronchoscopy;

2. Spiral CT: superior to incremental CT, the patient is shifted during the rotation of the tube-detector system. It enables to acquire large anatomical regions, like Thorax, in a single breath-hold. But still, the z-axis resolution is considerably worse than the in-plane resolution, and is a limiting factor for small bronchi.

3. Multi-slice CT: common systems image four slices at a time. This results in very low total acquisition time, but can also result in isotropic volume elements (voxels) in the final 3D dataset. Figure 9.8 displays a volume of interest of a classical multi-slice CT scanner of the lungs.

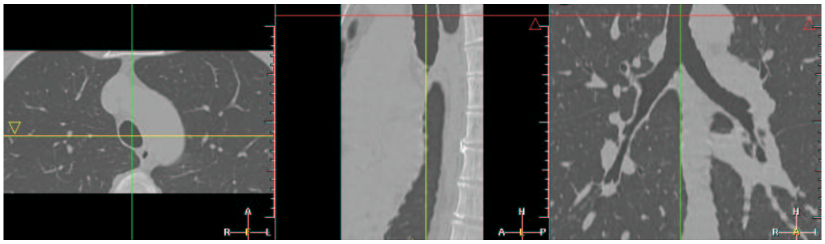


Figure 9.8. 3D Multi-Slice CT scanner of the Lungs: this is an isotropic volume of interest about 287 × 150 × 249 voxels.

Multi-slice CT is the ideal imaging modality for obtaining virtual bronchoscopic renderings of the smaller bronchi, since structures with diameter of 2mm should become clearly visible. However, the huge number of voxels can create a problem, as it overload the storage capacity of common workstations. Processing time in algorithms can also become so long than clinical application may not be realistic.

Segmentation techniques

They can be divided in three categories:

1. 2D methods: they can be completely manual, where the contours of the airway lumen is delineated manually in each axial images, but they strongly depend on the interobserver variability, and the loss of precision of the 2D segmentation towards the 3D information. They can also be semi-automatic, starting from an initial set of contours manually drawn, which is corrected and smoothed with detection algorithms, or by flooding gradient maps with region-growing in 2D [144]. However, those 2D methods work on the information contained in the cross-sections of the bronchi contains on not on the whole 3D data.

2. 3D methods: there is of course the usual set of 3D rendering techniques (**MPR** , **MIP** , volume or surface renderings). Other more elaborated methods are based on 3D region-growing algorithms. *Mori et al.* [124] detect an optimal threshold value in order to extract the airways, and once this segmentation is done, thin the airways to obtain the tree which is input in a recognition process for automatic labeling [123]. Those methods suffer from limitations due to the use of a threshold, and problems of accuracy for the small bronchi, since they provide a binary image as segmentation.

3. 2D to 3D methods: the principle is to segment the airways in each axial slice, and to reconstruct the 3D segmentation by combining the 2D segmentations, doing 3D/3D post-processing. Several techniques are based on the detection of the airways location in 2D [169]. But how important is the enhancement of the post-processing, the result depends on the 2D initialization, there is no pure 3D information involved in the detection. However, important improvements have been done in the field by *Fetita* [52], as well in 2D/3D as in pure 3D.

We have worked upon the use of our method involving both *Fast-Marching* and *Level-Sets* methods to extract the airways, and the airway tree. We encountered several problems due to the specificity of the bronchi in CT, during the initialization step. All tests are detailed in the following.

9.2.3 Applying our framework

Initialization: Region growing

Several problems exist with the use of the *Fast-Marching* algorithm as a region growing method for airway segmentation. Since it relies on the edge strength of the airway walls that weak at several places, the propagating front *leaks* out into the surrounding parenchyma. We have already seen that the *Fast-Marching* can flood into the surrounding pixels of the tubular structures, and we have built a method based on *Freezing* pixels (see section 8.3.1) in order to avoid *leakage*. This method was successfully applied for vascular tree extraction in chapter 8, using as speed function a multiscale filtering technique derived from work in [60]. But in the case of the lungs airway, the grey level information is very different, as shown in figure 9.9. In figure 9.9-right, we observe that the minima occur near the center of the airways and the maxima occur near the middle of the walls. But this information is relatively poor, and partial volume effects can occur: as the diameter of the airways decreases, partial volume averaging begins to increase the value within the lumen, and the *Fast-Marching* will flood the parenchyma at a weak wall, as shown in figure 9.10. The *Fast-Marching* algorithm is applied in this case with a potential based on the grey level information. Using $\mathcal{P}(\mathbf{x}) = \max\left(I(\mathbf{x}) - I_{airways}, 0\right) + w$, with $I_{airways}$ being an approximate value corresponding to air, cannot provide a result where the propagation critically depends on the weakness of the edges. Using the grey level information for the lungs is similar to using the gradient in the vascular contrast enhanced medical images: it is not valuable. The profile of the airway in figure 9.9-right is somehow similar to a "Mexican hat", but the Hessian information given by a measure based on its eigenvalues will detect the inner and outer walls of the airways, and will not give

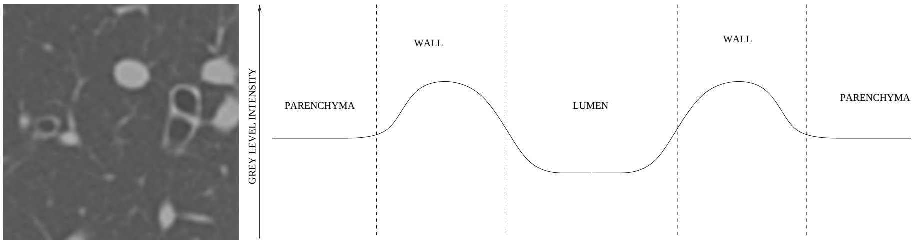


Figure 9.9. Profile of the airway grey level: Left image is a zoom on a 2D axial slice of the multi-slice reformat image of the lungs, we can distinguish the lumen in the black circle areas surrounded by the airway walls in bright intensities; right image is an illustration of the profile of the grey level intensity, along a line crossing through an airway center.

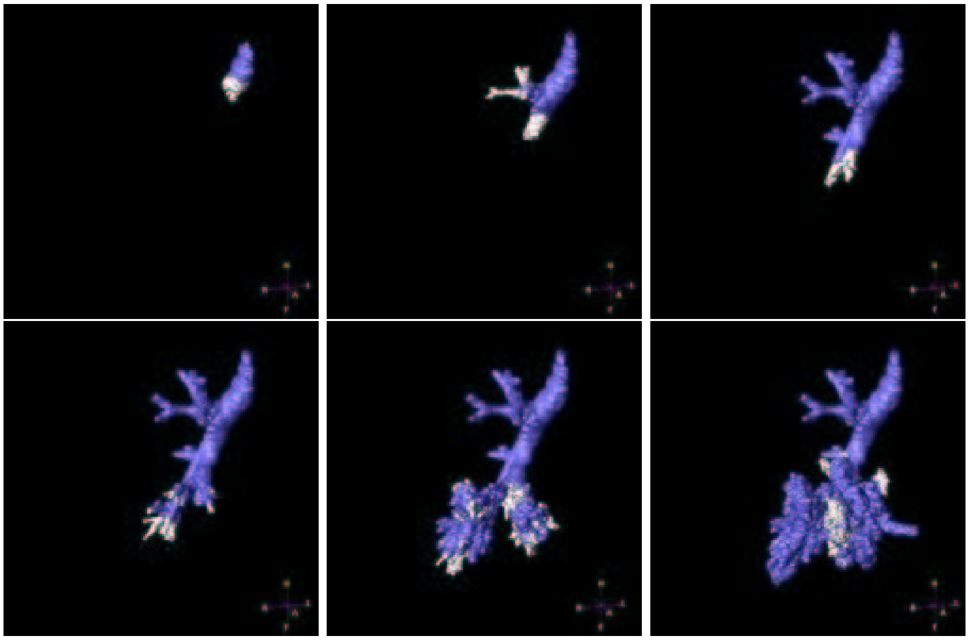


Figure 9.10. Flooding of the *Fast-Marching* in the parenchyma: the images are samples of the front propagation process in the lungs airways, with the *Freezing* methodology; the frozen pixels are represented in blue whereas the propagating parts are in white.

a high response in the lumen. Since then, this problem defines the limits of the use of the *Fast-Marching* in pre-segmentation, as well as with the use of a more complicated heuristic such as the *Freezing* algorithm.

However, there are other ways to create a pre-segmentation which can be input in a *Level-Sets* model. Among others, we decided to use methods developed by *Mori et al.* in [124]. The principle of this initialization is based on the reason of the failure of the *Fast-Marching* method: it focuses on the detection of the *flooding* inside the parenchyma. It extracts the inside of the area of the bronchus by tracing voxels with relatively small CT values corresponding to air without processing across voxels with relatively large CT values, assuming that the airway area is simply connected.

The method is a simple region-growing in the 3D image, starting from a point inside the trachea. This point, which will be the root of the final tree hierarchy can be easily detected in the 3D dataset, as shown in figure 9.8

The algorithm is the following:

Definition

- a start point $\mathbf{x}_0$: the region growing needs a seed point for starting. In order to set a protocol of segmentation, the start point needs to be always initialized in the same region, inside the trachea, before the first bifurcation in the tracheobronchial tree. Luckily enough, the trachea can be recognized stably in the multi-slice CT dataset.

Moreover this seed point will be the root of the tree hierarchy extracted at the end of the whole process, which enables to reduce user interaction to the setting of the seed point only.

- an initial threshold value I_0, a threshold value $I_{threshold}$ with a threshold step dI, for the original volume image I
- a segmentation defined by a binary mask $\mathcal{M}$ where $\mathcal{M}(\mathbf{x}) = 0$ if $\mathbf{x}$ is inside the object, and $\mathcal{M}(\mathbf{x}) = -1$ elsewhere.

Initialization

- $I_{threshold} = I_0$;
- $\mathcal{M}(\mathbf{x}_0) = 0$ and $\mathcal{M}(\mathbf{x}) = -1$ elsewhere;

Loop:

- at each iteration i, we binarize image I, defining the mask I_B where $I_B(\mathbf{x}) = 0$ if $I(\mathbf{x}) < I_{threshold}$, $I_B(\mathbf{x}) = -1$ elsewhere;
- we apply a connectivity algorithm, to connect to the pixels $\mathbf{x}$ that verify $\mathcal{M}(\mathbf{x}) = 0$ all voxels $\mathbf{y}$ with $I_B(\mathbf{y}) = 0$.
- For all voxels $\mathbf{y}$ connected, $\mathcal{M}(\mathbf{y}) = 0$;
- We count N_i the total number of voxels $\mathbf{x}$ with $\mathcal{M}(\mathbf{x}) = 0$ at iteration i (see figure 9.11);
- If $N_i > N_{max}$, the optimal threshold is $I_{threshold}$ and we stop;
- $I_{threshold} = I_{threshold} + dI$.

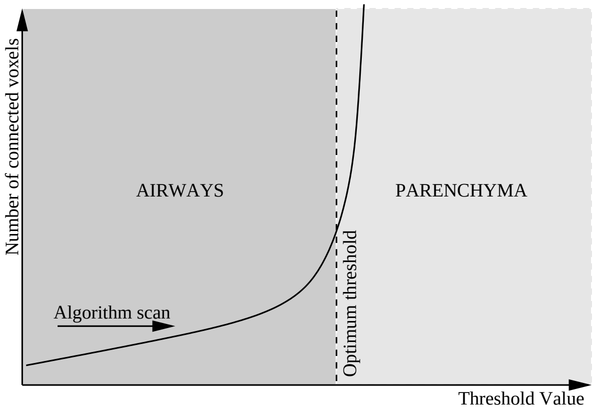


Figure 9.11. Detection of the optimal threshold value: When the threshold value is increased, the number of connected voxels below this threshold increases, until it reaches the optimal value; a superior threshold will lead to the flooding in the parenchyma.

When the number of segmented voxels exceeds the area threshold value N_{max}, the algorithm is stopped, just before explosion in the number of voxels visited occurs in figure 9.11.

Final Segmentation and Automatic Branch extraction

Since the initialization is obtained, the inside and outside regions are used in order to initialize the region-based forces of the *Level-Sets* model. In the case of the lungs, we used the sigmoidal region-based forces formulation for several reasons:

- the computing time is very important, considering the size of the dataset, and the use of constant forces across time reduces this cost;
- the parenchyma has a distribution very similar to that of the airways. The use of region-based forces should then induce the use of three different regions: one for the lumen, one for the parenchyma, and one for the soft tissues (and others). Managing three regions will greatly increase the computing time.
- if we use two regions instead of three, the parenchyma is contained in the outside region and reduces greatly the mean and increases the variance of the model. There is a huge risk that the outside has a variance too important, and shrinks the lumen segmented. This problem has been already presented in the application concerning the visualization of the colon polyps.

Several iterations are necessary in order to extract the lower bronchi, as shown in figure 9.12. Once this segmentation step is achieved, we extract the trajectories from the starting point $\mathbf{x}_0$, and we convert it into a tree, using the labeling methods of section 8.4. We optimize this extraction, in function of the length of the minimal branch to be extracted, as shown in figure 9.13, where we display the label map for several minimal length. Since this step represents a computing cost, this minimal length must be accurately set according to the needs of the clinician (since this length shrinks with the depth in the bronchial tree).

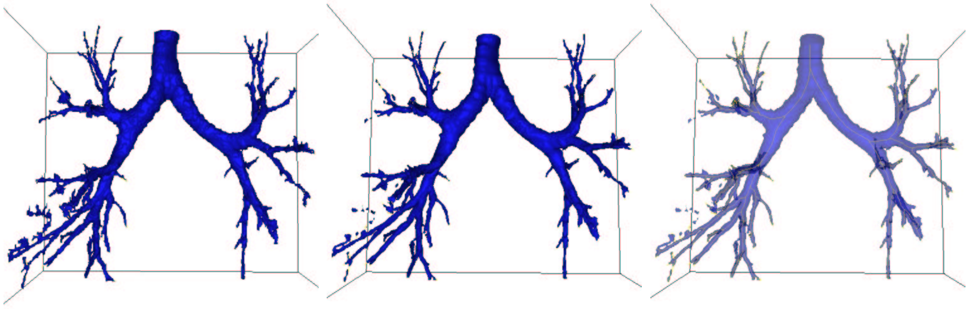


Figure 9.12. Three steps of the airways tree segmentation: Left image is the initialization given by the method [124] described in section 9.2.3; middle image is the surface of the airways after 40 iterations of the *Level-Sets* model; right image shows the whole tree extracted in the airways with the labeling algorithm illustrated in figure 9.13.

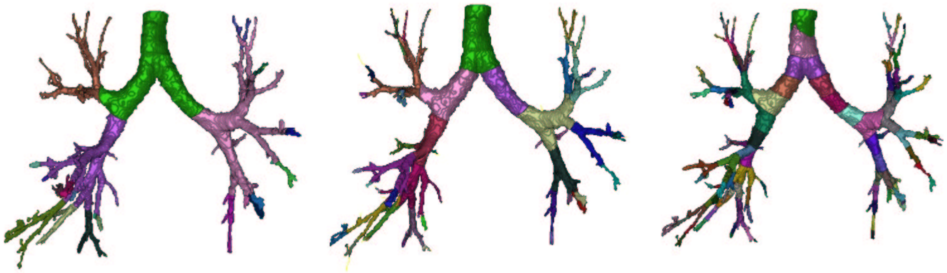


Figure 9.13. Labeling of the airways: Using the method described in section 8.4, we label the airways segmented, according to a chosen distance step; from left to right are shown the label maps for respective steps 100, 50 and 20.

9.2.4 Conclusion

Artifacts

Even with multi-slice technology, one problem remains unsolved: cardiac motion cannot be suppressed, despite fast acquisition. Only a trigger, as employed for cardiac imaging could provide assistance. This motion extends to the neighboring pulmonary parenchyma, and results in irregularities in the bronchial wall (see figure 9.14-left). The impact of this motion onto the segmentation is not clear, but it seems that the surface extracted near the hear in the lower left part of the bronchi is not correctly segmented (see figure 9.14-middle). These artifacts can be reduced by doing multi-slice CT acquisition with a trigger, as done for cardiac imaging, taking into account

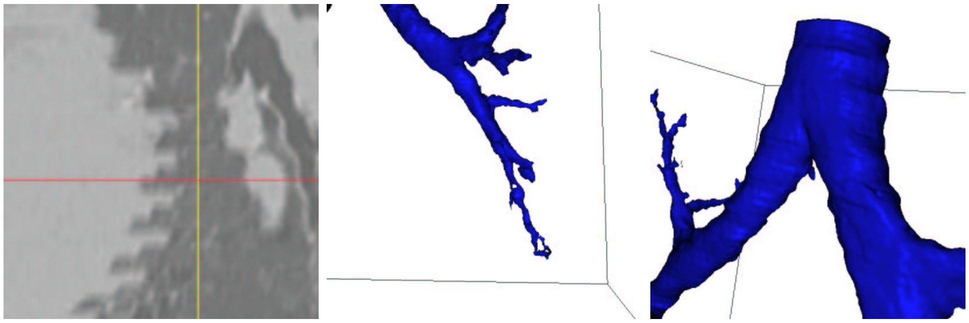


Figure 9.14. Two kind of artifacts in the multi-slice datasets: Left image shows the motion of the heart in a sagittal slice of the 3D dataset of figure 9.8 during acquisition; middle image shows the resulting poor segmentation of the lower-left part of the airways; right image displays the pulsation artifacts carried over from the aorta onto the trachea: these rings should not be confused with the natural tracheal cartilage.

cardiac pulsation for the time of acquisition.

Much more striking are artifacts produced by the aortic pulsations, transmitted to the trachea and the main bronchi: they appear as ring-like structures on the 3D surface, similarly to cartilage rings. However, they can easily be distinguished from the tracheal cartilages, since they appear horizontal in the lice direction (see figure 9.14-right). Those rings imply errors in the measures of the airways diameters, but there is no particular possibility to avoid them, since a correlation with the heart pulsations exists but is difficult to model.

Perspectives

Results are impressive, and we can easily obtain a virtual bronchoscopic view, using the tree structure to guide the virtual endoscope and the triangulated surface, obtained through th *Marching-Cubes* algorithm with the zero-level set of our level-set function (see figure 9.15)

However, there are several technical improvements that are currently missing

- Improving fast-marching for the airway initialization: the method developed has failed in giving an accurate initialization. The complex structure of the object, and the thin-walled bronchi plus the partial volume effect lead to wrong results. One possible extension could be to modify the speed function, adapting it to the depth of the current voxels involved in the computation, since the partial volume effect increases with the depth in the airway tree.

- Reducing the number of iterations of the *Level-Sets* model: 40 iterations is still a huge number if each iterations is computed on the whole volume.

- Improving the *Level-Sets* model: the smaller parts of the bronchi are not recovered, the region-based formulation is not dedicated to the extraction of the thin curves. Using co-dimension 2 geodesic contours, as done by *Lorigo et al.* [108] for vessels in MRA images, and using other expression of the flows like *Vasilevskiy and Siddiqi* [177].

- Developments: *Mori* was using its tree extraction method for automatic labeling of the bronchial tree [123]. If it is possible to assign the anatomical names to the bronchial branches extracted from CT images and to display the name of the currently observed branch on a virtual bronchoscopy image, it will help clinicians to understand the current observing position. This tree structure can also be input in a system to assist biopsies in the tracheobronchial tree, as done in [17].

- Validation: still, clinical evaluation of the method is not done, and in particular the evaluation of the tree structure extracted should be evaluated with the choice of the minimal length of the branches extracted.

- Benchmark: a possibility of a benchmark with the tremendous work of *Fetita* [52, 144] has been scheduled but is not achieved yet.

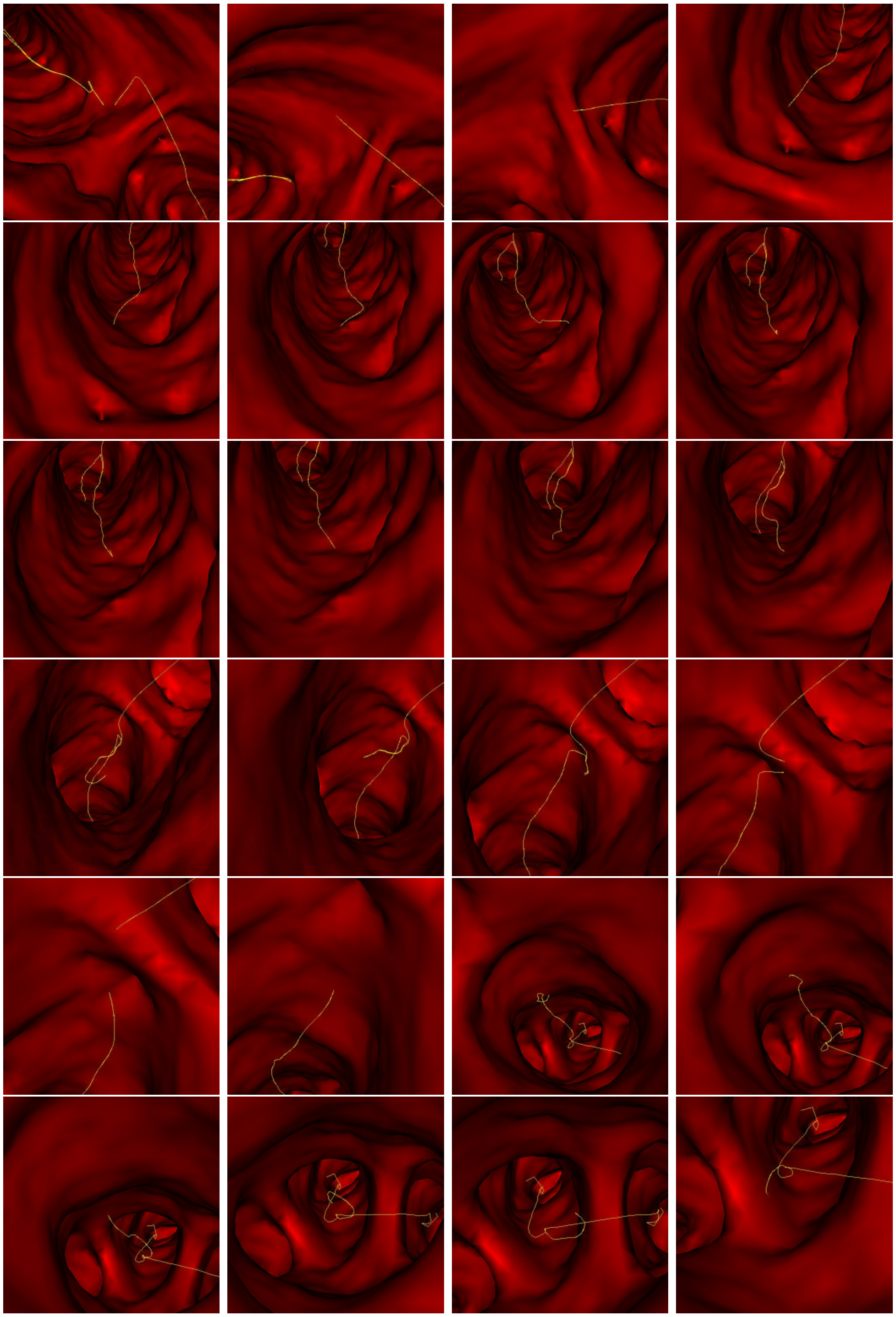


Figure 9.15. Virtual Bronchoscopy: the images are samples of a movie automatically generated with the surface and the tree extracted in the 3D dataset with our Path and Shape extraction framework.

9.3 Reconstruction of vessels in 2D and 3D images using *Perceptual Grouping*

Since their introduction, active contours [82] have been extensively used to find the contour of an object in an image through the minimization of an energy. In order to get a set of contours with T-junctions, we need many active contours to be initialized on the image. The level sets paradigm [113, 23] allows changes in topology. It enables to get multiple contours by starting with a single one. However, it does not give satisfying results when there are gaps in the data since the contour may propagate into a hole and then split into several curves where only one contour is desired. This is the problem encountered with *Perceptual Grouping* when we try to group a set of incomplete contours. For example, in a binary image like in figure 9.16 with a drawing of a shape with holes, human vision can easily fill in the missing boundaries and form complete curves. *Perceptual Grouping* is an old problem in computer vision. It has been approached more recently with energy methods [164, 72, 187]. These methods find a criteria for saliency of a curve component or for each point of the image. This saliency measure is based indirectly on a second order regularization snake-like energy ([82]) of a path containing the point. However, the final curves are generally obtained in a second step as ridge lines of the saliency criteria after thresholding. Motivated by this relationship between energy minimizing curves like snakes and completion contours, we worked upon finding a set of completion contours on an image as a set of energy minimizing curves.

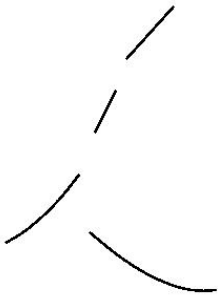


Figure 9.16. Examples of connected regions to be completed: The four regions are the four black components, on a bright background.

In order to solve global minimization for snakes, *Cohen and Kimmel* [34] used the minimal paths, as introduced in [87, 86]. The goal was to avoid local minima without demanding too much on user initialization, which is a main drawback of classic snakes [29]. Only two end points were needed. The numerical method has the advantage of being consistent (see [34]) and efficient using the Fast Marching algorithm introduced in [161]. In [31], the author proposed a way to use this minimal path approach to find a set of curves drawn from a set of points in the image. We also introduced a technique that automatically finds a set of *key* end points. In this chapter, we extend the previous approach to connected components instead of end points. In order to obtain a set of most salient contour curves, we find a set of minimal paths between pairs of connected components.

This approach is then extended for application in the completion of tube-like structures in 2D and 3D images. The problem is here to complete a partially detected object, based on some detected connected components that belong to this object.

For *Perceptual Grouping* , the potential P to be minimized along the curves is usually an image of edge points that represent simple incomplete shapes, as in figure 9.16. These edge points are represented as a binary image with small potential values along the edges and high values at the background. The potential could also be defined as edges weighted by the value of the gradient or as a function of an estimate of the gradient of the image itself, $P = g(\|\nabla I\|)$, like in classic snakes. The potential could also be a grey level image as in [34]. It could also be a more complicated function of the grey level. In our real examples of vascular structures in 2D and 3D, we use a potential based on a vesselness filter [60].

We present in Section 9.3.1 how to find a set of curves from a given set of unstructured points. Grouping the points in connected components, we propose a way to find the pairs of linked connected components and the paths between them. We then extend this approach to 3D and show an application in 3D medical images.

9.3.1 Finding Contours from a Set of Connected Components

Minimal Path between two Regions

The method of [34], detailed in the previous section allows to find a minimal path between two endpoints. This is a straightforward extension to define a minimal path between two regions of the image. Given two connected regions of the image $\mathbf{R}_0$ and $\mathbf{R}_1$, we consider $\mathbf{R}_0$ as the starting region and $\mathbf{R}_1$ as a set of end points. The problem is then finding a path minimizing energy among all paths with start point in $\mathbf{R}_0$ and end point in $\mathbf{R}_1$. The minimal action is now defined by

$$U(p) = \inf_{\mathcal{A}_{\mathbf{R}_0,p}} E(C) = \inf_{p_0 \in \mathbf{R}_0} \inf_{\mathcal{A}_{p_0,p}} E(C) \tag{9.3}$$

where $\mathcal{A}_{\mathbf{R}_0,p}$ is the set of all paths starting at a point of $\mathbf{R}_0$ and ending at p. This minimal action can be computed the same way as before in table 2.1, with the alive set initialized as the whole set of points of $\mathbf{R}_0$, with $U = 0$ and trial points being the set of 4-connexity neighbors of points of $\mathbf{R}_0$ that are not in $\mathbf{R}_0$. Back-propagation by gradient descent on U from any point p in the image will give the minimal path that join this point with region $\mathbf{R}_0$.

In order to find a minimal path between region $\mathbf{R}_1$ and region $\mathbf{R}_0$, we determine a point $p_1 \in \mathbf{R}_1$ such that $U(p_1) = \min_{p \in \mathbf{R}_1} U(p)$. We then back-propagate from p_1 to $\mathbf{R}_0$ to find the minimal path between p_1 and $\mathbf{R}_0$, which is also a minimal path between $\mathbf{R}_1$ and $\mathbf{R}_0$.

Minimal Paths from a Set of Connected Components

We are now interested in finding many or all contours in an image. We assume that from some preprocessing, or as data, we have an initial set of contours. We denote $\mathbf{R}_k$ the connected components of these contours. We propose to find the contours as a set of minimal paths that link pairs of regions among the $\mathbf{R}_k$'s. If we also know which pairs of regions have to be linked together, finding the whole set of contours is a trivial application of the previous section. The problem we are interested in here is also to find out which pairs of regions have to be connected by a contour. Since the set of contours $\mathbf{R}_k$'s is assumed to be given unstructured, we do not know in advance how the regions connect. This is the key problem that is solved here using a minimal action map.

Method

Our approach is similar to computing the distance map to a set of regions and their Voronoi diagram. However, we use here a weighted distance defined through the potential P. This distance is obtained as the minimal action with respect to P with zero value at all points of regions $\mathbf{R}_k$. Instead of computing a minimal action map for each pair of regions, as in Section 9.3.1, we only need to compute one minimal action map in order to find all paths. At the same time the action map is computed we determine the pairs of regions that have to be linked together. This is based on finding meeting points of the propagation fronts. These are *saddle points* of the minimal action U. These saddle points were already used for closed boundary extraction in [34] In Section 1.1.2, we said that calculation of the minimal action can be seen as the propagation of a front through equation (1.5). Although the minimal action is computed using fast marching, the level sets of U give the evolution of the front. During the fast marching algorithm, the boundary of the set of alive points also gives the position of the front. In the previous section, we had only one front evolving from the starting region $\mathbf{R}_0$. Since all points p of regions $\mathbf{R}_k$ are set with $\mathcal{U}(p) = 0$, we now have one front evolving from each of the starting regions $\mathbf{R}_k$. In what follows when we talk about front meeting, we mean either the geometric point where the two fronts coming from different $\mathbf{R}_k$'s meet, or in the discrete algorithm the first alive point which connects two components from different $\mathbf{R}_k$'s (see Figures 9.17 and 9.18).

We use the fact that given two regions $\mathbf{R}_1$ and $\mathbf{R}_2$, the saddle point $\mathbf{s}$ where the two fronts starting from each region meet can be used to find the minimal path between $\mathbf{R}_1$ and $\mathbf{R}_2$. Indeed, the minimal path between the two regions has to pass by the meeting point $\mathbf{s}$. This point is the point half way (in energy) on a minimal path between $\mathbf{R}_1$ and $\mathbf{R}_2$. Back-propagating from $\mathbf{s}$ to $\mathbf{R}_1$ and then from $\mathbf{s}$ to $\mathbf{R}_2$ gives the two halves of the path.

Notations and definitions

Here are some definitions that will be used in what follows. X being a set of points in the image, U_X is the minimal action obtained by Fast Marching with potential $\tilde{P}$ and starting points $\{p, p \in X\}$. This means that all points of X are initialized as

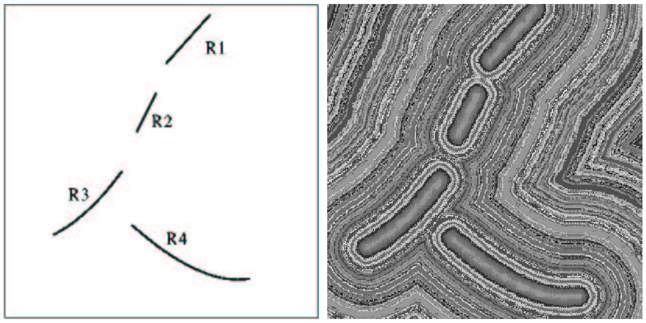


Figure 9.17. Minimal Action map from the four regions of the example of figure 9.16: On the right with a random LUT to show the level sets.

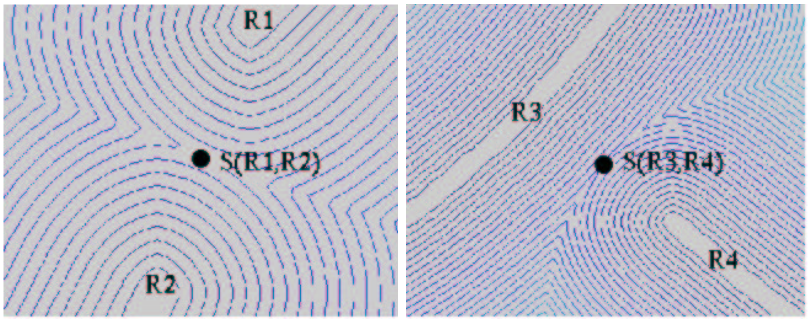


Figure 9.18. Zoom on *saddle points* **between regions:** Left image shows the iso-action levels near the saddle point between $\mathbf{R}_1$ and $\mathbf{R}_2$

alive points with value 0. All their 4-connexity neighbors that are not in X are *trial* points. This is easy to see that $U_X = \min_{p \in X} U_p$. X may be a connected component R or a set of connected components.

The *label* l at a point p is equal to the index k of the region $\mathbf{R}_k$ for p closer in energy to $\mathbf{R}_k$ than to other regions $\mathbf{R}_j$. This means that minimal action $U_{\mathbf{R}_k}(p) \leq U_{\mathbf{R}_j}(p), \forall j \neq k$. We define the region $L_k = \{p / l(p) = k\}$. If $X = \cup_j \mathbf{R}_j$, we have $U_X = U_{\mathbf{R}_k}$ on L_k and the computation of U_X is the same as the simultaneous computation of each $U_{\mathbf{R}_k}$ on each region L_k. These are the simultaneous fronts starting from each $\mathbf{R}_k$.

A *saddle point* $\mathbf{s}(\mathbf{R}_i, \mathbf{R}_j)$ between $\mathbf{R}_i$ and $\mathbf{R}_j$ is the first point where the front starting from $\mathbf{R}_i$ to compute $\mathcal{U}_{\mathbf{R}_i}$ meets the front starting from $\mathbf{R}_j$ to compute $\mathcal{U}_{\mathbf{R}_j}$; At this point, $U_{\mathbf{R}_i}$ and $U_{\mathbf{R}_j}$ are equal and this is the smallest value for which they are equal.

Two different regions among the $\mathbf{R}_k$'s will be called *linked regions* if they are selected to be linked together. The way we choose to link two regions is to select some *saddle points*. Thus regions $\mathbf{R}_i$ and $\mathbf{R}_j$ are *linked regions* if their *saddle point* is among the selected ones.

A *cycle* is a sequence of different regions $\mathbf{R}_k, 1 \leq k \leq K$, such that for $1 \leq k \leq K-1$, $\mathbf{R}_k$ and $\mathbf{R}_{k+1}$ are *linked regions* and $\mathbf{R}_K$ and $\mathbf{R}_1$ are also *linked regions*.

Finding and Selecting Saddle Points

The main goal of our method is to obtain all significant paths joining the given regions. However, each region should not be connected to all other regions, but only to those that are closer to them in the energy sense. There are many possibilities for deciding which regions connect together depending on the kind of data and application. In some cases, the goal would be to detect closed curves and avoid forming branches, as in [31]. Then the criterion would be to constrain a region to be linked to at most two other regions in order to make *cycles*. In our context, we are interested in detecting branches and avoiding closed curves. Therefore the criterion for two regions $\mathbf{R}_i$ and $\mathbf{R}_j$ to be connected is that their fronts meet without creating a "cycle".

We see in Figure 9.18 a zoom on the saddle points detected between regions $\mathbf{R}_1$ and $\mathbf{R}_2$ and $\mathbf{R}_3$ and $\mathbf{R}_4$. Once a *saddle point* $\mathbf{s}(\mathbf{R}_i, \mathbf{R}_j)$ is found and selected, back-propagation relatively to final energy U should be done both ways to $\mathbf{R}_i$ and to $\mathbf{R}_j$ to find the two halves of the path between them. We see in Figure 9.19 this back-propagation at each of the three automatically selected *saddle points*. They link $\mathbf{R}_1$ to $\mathbf{R}_2$, $\mathbf{R}_2$ to $\mathbf{R}_3$ and $\mathbf{R}_3$ to $\mathbf{R}_4$. At a saddle point, the gradient is zero, but the direction of descent towards each point are opposite. For each back-propagation, the direction of descent is the one relative to each region. This means that in order to estimate the gradient direction toward $\mathbf{R}_i$, all points in a region different from L_i have their energy put artificially to ∞. This allows finding the good direction for the gradient descent towards $\mathbf{R}_i$. However, as mentioned earlier, these back-propagations have to be done only for selected *saddle points*. In the fast marching algorithm we have a simple way to find *saddle points* and update the *linked regions*.

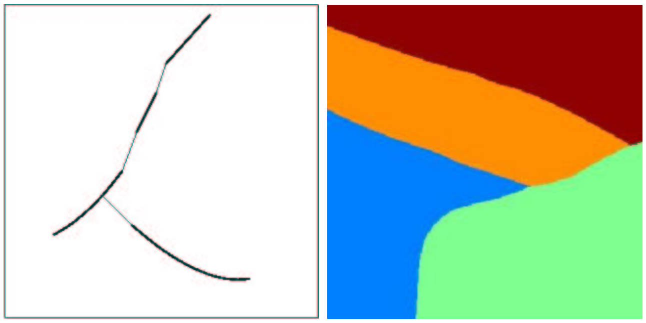


Figure 9.19. Example with four regions: On the left we show the minimal paths obtained by back-propagation from the three *saddle points* to each of the regions from where the front comes; on the right, and the Voronoi diagram obtained.

As defined above, the region L_k associated with a region $\mathbf{R}_k$ is the set of points p of the image such that minimal energy $U_{\mathbf{R}_k}(p)$ to $\mathbf{R}_k$ is smaller than all the $\mathcal{U}_{\mathbf{R}_j}(p)$ to other regions $\mathbf{R}_j$. The set of such regions L_k covers the whole image, and forms the Voronoi diagram of the image (see figure 9.19). All *saddle points* are at a boundary between two regions L_k. For a point p on the boundary between L_j and L_k, we have $\mathcal{U}_{\mathbf{R}_k}(p) = U_{\mathbf{R}_j}(p)$. The *saddle point* $\mathbf{s}(\mathbf{R}_k, \mathbf{R}_j)$ is a point on this boundary with minimal value of $U_{\mathbf{R}_k}(p) = U_{\mathbf{R}_j}(p)$. This gives us a rule to find the *saddle points* during the fast marching algorithm.

Each time two fronts coming from $\mathbf{R}_k$ and $\mathbf{R}_j$ meet for the first time, we define the meeting point as $\mathbf{s}(\mathbf{R}_k, \mathbf{R}_j)$. This means that we need to know for each point of the image from where it comes. This is easy to keep track of its origin by generating an index map updated at each time a point is set as alive in the algorithm. Each point of region $\mathbf{R}_k$ starts with label k. Each time a point is set as alive, it gets the same label as the points it was computed from in formula (1.7). In that formula, the computation of $U_{i,j}$ depends only on at most two of the four pixels involved. These two pixels, said A_1 and B_1, have to be with the same *label*, except if (i, j) is on the boundary between two labels. If A_1 and B_1 are both alive and with different labels k and l, this means that regions $\mathbf{R}_k$ and $\mathbf{R}_l$ meet there. If this happens for the first time, the current point is set as the *saddle point* $\mathbf{s}(\mathbf{R}_k, \mathbf{R}_l)$ between these regions. A point on the boundary between $\mathbf{R}_k$ and $\mathbf{R}_l$ is given the label of the neighbor point with smaller action A_1. At the boundary between two labels there can be a slight error on labeling. This error of at most one pixel is not important in our context and could be refined if necessary.

Algorithm

The algorithm for this section is described in Table 9.1 and illustrated in figures 9.17 to 9.19. When there is a large number of $\mathbf{R}_k$'s, this does not change much the computation time of the minimal action map, but this makes more complex dealing with the list of linked regions and *saddle points* and testing for *cycles*.

The way we chose to test for cycles is as follows. Assume a saddle point between regions $\mathbf{R}_i$ and $\mathbf{R}_j$ is found. We then test if there is already a link between these regions through other regions. This means we are looking for a sequence of different regions $\mathbf{R}_k, 1 \leq k \leq K$, with $\mathbf{R}_1 = \mathbf{R}_i$ and $\mathbf{R}_K = \mathbf{R}_j$, such that for $1 \leq k \leq K-1$, $\mathbf{R}_k$ and $\mathbf{R}_{k+1}$ are *linked regions*.

This kind of condition can be easily implemented using a recursive algorithm. When two regions $\mathbf{R}_i$ and $\mathbf{R}_j$ are willing to be connected - i.e. that their fronts meet - a table storing the connectivity between each region enables to detect if a link already exists between those regions. Having N different regions, we fill a matrix $M(N, N)$ with zeros, and each time two regions $\mathbf{R}_i$ and $\mathbf{R}_j$ meet without creating a cycle, we set $M(i, j) = M(j, i) = 1$. Thus, when two regions meet, we apply the algorithm detailed in table 9.2.

If two regions are already linked, the pixel where their fronts meet is not considered as a valuable candidate for back-propagation. The algorithm stops automatically when all regions are connected.

Minimal paths between Regions $\mathbf{R}_k$

- Initialization:
  - $\mathbf{R}_k$'s are given
  - $\forall k, \forall p \in \mathbf{R}_k, V(p) = 0; l(p) = k;$ p *alive*.
  - $\forall p \notin \cup_k \mathbf{R}_k, V(p) = \infty; l(p) = -1;$ p is *far* except 4-connexity neighbors of $\mathbf{R}_k$'s that are *trial* with estimate U using equation (1.7).
- Loop for computing $V = U_{\cup_k \mathbf{R}_k}$:
  - Let $p = (i_{min}, j_{min})$ be the *Trial* point with the smallest action U;
  - Move it from the *Trial* to the *Alive* set with V(p) = U(p);
  - Update $l(p)$ with the same index as point A_1 in formula (1.7). If $R(A_1) \neq R(B_1)$ and we are in case 1 in table 2.2 where both points are used and if this is the first time regions of labels $l(A_1)$ and $l(B_1)$ meet, $\mathbf{s}(\mathbf{R}_{l(A_1)}, \mathbf{R}_{l(B_1)}) = p$ is set as a *saddle point* between $\mathbf{R}_{l(A_1)}$ and $\mathbf{R}_{l(B_1)}$. If adding a link between these regions does not create a *cycle*, they are set as *linked regions* and $\mathbf{s}(\mathbf{R}_{l(A_1)}, \mathbf{R}_{l(B_1)}) = p$ is selected,
    For each neighbor (i, j) of (i_{min}, j_{min}):
    - If (i, j) is *Far*, add it to the *Trial* set;
    - If (i, j) is *Trial*, update action $U_{i,j}$.
- Obtain all paths between selected *linked regions* by back-propagation each way from their *saddle point* (see Section 9.3.1).

Table 9.1. Algorithm of Section 9.3.1

Application

The method can be applied to connected components from a whole set of edge points or points obtained through a preprocessing. Finding all paths from a given set of points is interesting in the case of a binary potential defined, like in Figure 9.17, for *Perceptual Grouping* . It can be used as well when a special preprocessing is possible, either on the image itself to extract characteristic points or on the geometry of the initial set of points to choose more relevant points. We show in figures 9.20 and 9.21 an example of application to a medical image of the hip where the objects of interest are the vessels. Potential P is defined using ideas from [60] on vesselness filter (detailed later in section 9.3.2). About vessel detection, see also [177, 107].

9.3.2 Finding a Set of Paths in a 3D Image

Extension to 3D

We now extend our approach to finding a set of 3D minimal paths between regions in 3D images. All definitions and algorithms of section 9.3.1 are not affected by changing the dimension of the image from 2D to 3D. The main changes are that 4-connexity in 2D is now 6-connexity in 3D and that we deal with minimal paths and minimal action in 3D images (see section 2.1 for the 3D extension of the the fast marching).

```
Algorithm for Cycle detection when a region $\mathbf{R}_i$ meets a region $\mathbf{R}_j$:
$Test(i, j, M, i); with$
$Test(i, j, M, l)$;
  • if $M(l, j) = 1$, return 1;
  • else
      – count=0;
      – for $k \in [1, N]$ with $k \neq i, k \neq j, k \neq l$ : count $+=$ $Test(k, j, M, l)$;
      – return count;
```

Table 9.2. Cycle detection

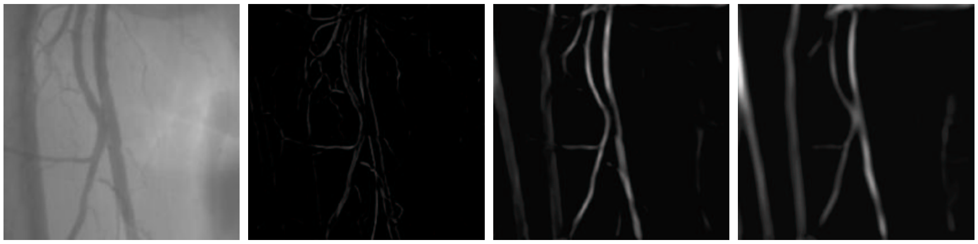


Figure 9.20. Multiscale vessel enhancement: First image is the original dataset; All other images are from left to right the filter response with respective kernels 1, 3, and 5.

Application to Real Datasets: a MR Image of the Aorta

The problem here is to complete a partially detected object. In figure 3.12 is shown a 3D MR dataset of the aorta, which presents a typical pathology: an abdominal aortic aneurysm. The anatomical object is made visible on the image by injecting a contrast product before the image acquisition.

We propose here to give a method for extracting from the grey level image a set of paths that will represent an approximate skeleton of the tree structure. This is based on extracting first a set of unstructured voxels or regions that belong to the object. Notice that [177, 107] give different methods to detect vessels but ours is much simpler and faster.

For this, we propose to extract valuable information from this dataset, computing a multi-scale vessel enhancement measure, based on the work of [60] on ridge filters. Having extracted the three eigenvalues of the Hessian matrix computed at scale σ, ordered $|\lambda_1| \leq |\lambda_2| \leq |\lambda_3|$, we define the vesselness function as done in the preceding chapter 8.

In figure 9.22 you can observe the response of the filter, based on the Hessian information, at three different scales: $\sigma = 1, 5, 10$. Visualization is made with Maximum Intensity Projection (MIP). Using this information computed at several scales, we can take as potential the maximum of the response of the filter across all scales (Fig. 9.23-left). And we can easily give a very constrained threshold of this image,

Figure 9.21. ***Perceptual Grouping* on a 2D Medical Image**: left image is the vesselness potential; middle and right images show that from the set of regions obtained from thresholding of potential image, our method finds links between these regions as minimal paths with respect to the potential.

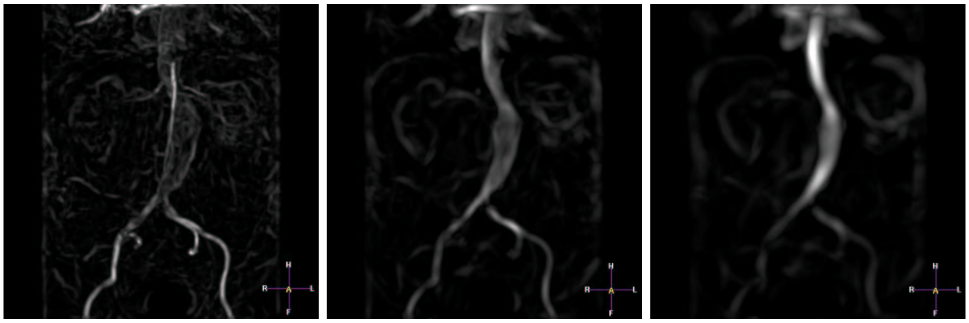


Figure 9.22. Ridge detection at three different scales: $\sigma = 1, 5, 10$ (MIP visualization of the 3D images)

that will lead to sets of unstructured voxels that surely belong to the anatomical object of interest, as shown in figure 9.23-middle.

Based on this set of regions, we apply our algorithm of section 9.3.1, using the 3D version of the Fast-Marching algorithm presented in section 2.1. We find the set of paths that connect altogether all the seed regions in our image, leading to the representation shown in figure 9.23-right.

9.3.3 Conclusion

We presented a new method that finds a set of contour curves in an image. It was applied to *Perceptual Grouping* to get complete curves from a set of edge regions with gaps. The technique is based on finding minimal paths between two end points [34]. However, in our approach, start and end points are not required as initialization. Given a unstructured set of regions, the pairs of regions that had to be linked by

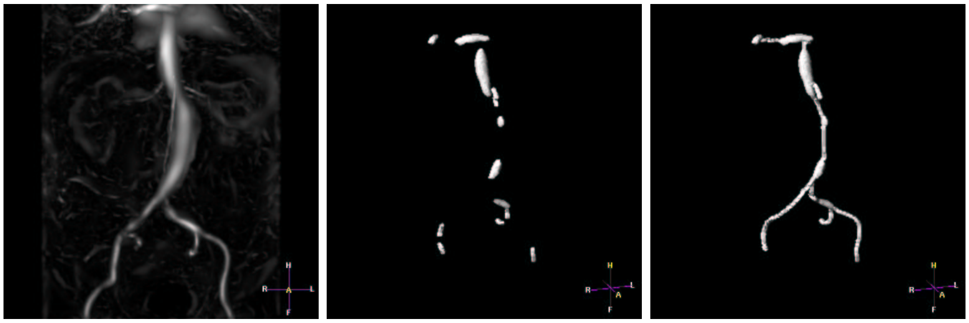


Figure 9.23. ***Perceptual Grouping*** **in the aorta of figure 3.12:** from left to right, visualization of the 3D potential (MIP view) obtained from the different scale of previous figure; a rough detection of the aorta; the Reconstructed aorta.

minimal paths are automatically found. Once *saddle points* between pairs of regions are found, paths are drawn on the image from the selected *saddle points* to both points of each pair. This gives the minimal paths between selected pairs of regions. The whole set of paths completes the initial set of contours and allows to close these contours. We applied this method in order to reconstruct vascular structures, and we showed examples for 2D vascular image and 3D medical dataset of the aorta. In case a refinement is needed, this method could be an efficient way to initialize geodesic contours. Other developments could lead to applications in roads detection in aerial images [63].

Conclusion

Résumé et contributions

Dans ce manuscrit, nous nous sommes intéressés à l'étude des courbes et surfaces en imagerie médicale tridimensionnelle.

Extraction de chemins

Nous avons développé un ensemble de techniques originales, basées sur les travaux préliminaires de *Cohen et Kimmel* [34], pour étendre l'extraction de chemins aux images tridimensionnelles. Nous nous sommes aussi attachés à fournir un éventail d'implémentations différentes afin de réduire les temps de calculs nécessaires, et de réduire l'intéraction d'un éventuel utilisateur.

Ces résultats ont mené à diverses applications, dont les plus avancées concernent en premier lieu l'implémentation d'un système de tracé automatique de chemins pour l'endoscopie virtuelle, qui a fait par la suite l'objet d'une validation clinique et d'une application industrielle, puisque ce système est dorénavant intégré dans un produit commecial, une console de traitement d'images médicales, *EasyVision*, développée par Philips Medical Systems. La seconde application est la construction d'un outil de délinéation intéractive des contours d'un objet dans des images bidimensionnelles, sur la base du modèle du *Live-Wire* [49, 127]. L'objéctif était de fournir un outil de tracé de contours semi-automatique, à l'aide des méthodes de chemins minimaux. Le résultat intègre une fonctionnalité intéressante au sens que l'utilisateur peut "apprendre" au programme quels sont les contours de l'image qu'il recherche. Cette méthode a fourni des résultats prometteurs, est fera sans doute l'objet d'une intégration à un environnement de traitement d'image, pour les contours des ventricules du coeur en ultrasons.

Extraction de surfaces

Dans un deuxième temps nous nous sommes intéressés à l'extraction de surfaces à l'aide des algorithmes de calcul de chemins minimaux. Nous avons montré le lien avec des techniques similaires en morphologie mathématique, notamment la *Ligne de partage des eaux* [180], et nous avons montré l'intéret d'une telle méthode, qui est rapide mais approximative, pour initialiser des algorithmes plus complexes, plus "savants", mais qui ont des temps de calculs beaucoup trop longs, comme les *Ensembles* *de Niveaux*. Nous avons présenté une méthode regrouppant ces diverses techniques en un seul et même algorithme.

Cet algorithme a ensuite été appliqué à des problèmes de segmentation et de visualisation difficiles, au sens de la topologie des objets à récupérer. En premier lieu, nous avons extrait des anévrismes cérébraux, qui sont des gonflements sur des veines du cerveau dont la rupture peut entrainer une hemorragie cérébrale fatale. Ces anévrismes ont une grande variété de formes et le modèle adéquat de segmentation ne doit pas présenter d'à priori sur la structure de l'objet à segmenter. Nous avons ensuite le même principe à la segmentation et la visualisation de polypes du colon, où nous avons présenté une méthode originale de discrimination des zones à observer, en utilisant les propriétés de la surface obtenue (principalement sa courbure). Cette dernière méthode est à l'origine de développements plus poussés pour la détection automatique des polypes du colon, qui vient de commencer à Philips Medical Systems.

Extraction de structures arborescentes

Finalement dans la dernière partie de la thèse nous nous sommes attachés à adapter nos algorithmes au cas particulier des structures arborescentes, où l'extraction de chemins et de surfaces trouve une utilisation originale. Nous avons commencé par développer une technique de segmentation rapide, avec une initialisation très limitée (à un point), adaptée aux structures tubulaires, sans aucune contrainte sur la topologie de l'objet final. Nous avons ensuite fourni un moyen d'obtenir une segmentation de précision sous-voxélique utilisant le premier algorithme comme initialisation.

De plus, pour offrir la possibilité d'une analyse complète d'un réseau arborescent, il faut pouvoir fournir les outils pour d'une part une navigation intelligente dans des données 3D, et d'autre part un moyen d'indexation pour le repérage au sein de cette structure. Nous avons tout d'abord élargi les capacités de notre système d'extraction de trajectoires à l'extraction d'un ensemble de trajectoires, puis au cas de l'extraction d'arborescences.

Ces techniques ont ensuite été appliquées à la segmentation et la reconstruction de réseaux vasculaires et artériels, dans des images de produit de contraste tridimensionnelles. Comparant les résultats obtenus à l'état de l'art dans ce domaine, nous concluons de la validité de notre méthode qui donne avantageusement une précision sous-voxélique des surfaces de nos objets tubulaires, en des temps intéractifs. L'utilisation de l'information de structure arborescente nous permet de localiser l'information de section des objets qui nous permet clairement de distinguer les pathologies dont ils font l'objet, comme les sténoses et les anévrismes.

Problèmes rencontrés et Perspectives

Il reste malgré tout que les techniques d'extraction d'arborescences n'ont pas fait l'objet à l'heure actuelle de validation clinique. L'exploitation des résultats montrent que la méthode d'initialisation, si elle est très rapide, ne donne pas totalement satisfaction au sens qu'elle peut ne pas récupérer des branches de nos structures qui sont très fines, auquel cas, la méthode de reconstruction de la section 9.3 peut s'avérer intéressante. Mais ce développement n'a pas été fait à l'heure actuelle.

Donc les perspectives de ce travail portent essentiellement sur une étape de validation clinique, comme cela a pu être fait dans le cas de l'endoscopie virtuelle. Quoi qu'il en soit chacune des applications présentées à chaque partie donne la direction pour de possibles développements, comme la visualisation de polypes, et l'extraction de structures arborescentes.

Mais les méthodes mathématiques mises en oeuvre pour l'extraction de chemins, et l'extraction de surfaces que nous avons développées dans ce manuscrit peuvent être utilisées dans un cadre beaucoup plus général que l'imagerie médicale et donner lieu à des applications dans d'autres cadres industriels.

Bibliography

[1] D. Adalsteinsson, R. Kimmel, R. Malladi, J.A. Sethian, “Center for Pure and Applied Mathematics, Univ. of California, Berkeley, CA 94720”, Tech. Rep., Fast Marching Methods for Computing Solutions to Static Hamilton-Jacobi Equations, 1996.

[2] D. Adalsteinsson & J.A. Sethian, “A Fast Level Set Method for Propagating Interfaces”, *Journal of Computational Physics*, vol. 118, pp. 269–277, 1995.

[3] D. Adalsteinsson & J.A. Sethian, “The Fast Construction of Extension Velocities in Level Set Methods”, *Journal of Computational Physics*, vol. 148, pp. 2–22, 1999.

[4] A.A. Amini, T.E. Weymouth, R.C. Jain, “Using Dynamic Programming for Solving Variational Problems in Vision”, *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 12, no. 9, pp. 855–867, September 1990.

[5] F. Angella, *Modeles Deformables et Systemes Particulaires - Application a l’Extraction de Structures Arborescentes en Analyse d’Images*, Ph.D. thesis, Université Bordeaux I, 2001.

[6] F. Angella, O. Lavialle, P Baylou, “A deformable and Expansible Tree for Structure Recovery”, in *Proceedings of the IEEE International Conference on Image Processing (ICIP’96)*, vol. 1, pp. 241–245, 1996.

[7] C. Baillard, P. Hellier, C. Barillot, “Segmentation of brain 3D MR images using level sets and dense registration”, *Medical Image Analysis*, vol. 5, no. 3, pp. 185–194, September 2001.

[8] F. Barbaresco & B. Monnier, “Minimal Geodesics Bundles by Active Contours : Radar Application for Computation of Most Threatening Trajectories Areas and Corridors”, in *Proceedings of European Signal and Image Processing Conference, EUSIPCO’00*, Tampere, Finland, sept 2000.

[9] E. Bardinet, *Constrained deformable models - Applications to cardiac imagery*, Ph.D. thesis, University of Paris IX-Dauphine, 1995.

[10] W. Barrett & E. Mortensen, “Interactive live-wire boundary extraction”, *Medical Image Analysis*, vol. 1, no. 4, pp. 331–341, 1997.

[11] R. Bellman & R. Kalaba, *Dynamic Programming and modern control theory*, London mathematical society monographs, London, 1965.

[12] M. Bertalmio, G. Sapiro, G. Randall, “Region Tracking on Level-Sets Methods”, in *Scale Space Theories in Computer Vision - Scale Space’99*, pp. 330–338, 1999.

[13] J. Bloomenthal, *Introduction to Implicit Surfaces*, Morgan Kaufmann Publishers, San Francisco, California, 1997.

[14] J. Bloomenthal & K. Shoemake, “Convolution surfaces”, *Computer Graphics*, vol. 25, no. 4, pp. 251–256, July 1991.

[15] H. Blum, “A transformation for extracting new descriptors of shape”, in *Models for the Perception of Speech and Visual Forms*, W. Wathen-Dunn (ed.), MIT Press, Amsterdam, pp. 362–380, 1967.

[16] G. Borgefors, “Distance Transformations in Arbitrary Dimensions”, *Computer Vision, Graphics, and Image Processing: Image Understanding*, vol. 27, no. 3, pp. 321–345, September 1984.

[17] I. Bricault, G. Ferretti, P. Cinquin, "Multi-level Strategy for Computer-Assisted Transbronchial Biopsy", in *Proceedings of the first International Conference on Medical Image Computing and Computer-Assisted Intervention, (MICCAI'98)*, 1998.

[18] D. Buthiau, E. Antoine, J.C. Piette, D. Nizri, P. Baldeyrou, D. Khayat, "Virtual tracheobronchial endoscopy: educational and diagnostic value", *Surg Radiol Anat*, vol. 18, no. 2, pp. 125–131, 1996.

[19] M.P. Cani & M. Desbrun, "Animation of Deformable Models Using Implicit Surfaces", *IEEE Transactions on Visualization and Computer Graphics*, vol. 3, no. 1, March 1997.

[20] J. Canny, "A Computational Approach to Edge Detection", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 8, no. 6, pp. 679–698, November 1986.

[21] V. Caselles, F. Catté, T. Coll, F. Dibos, "A Geometric Model for Active Contours", *Numerische Mathematik*, vol. 66, pp. 1–31, 1993.

[22] V. Caselles, R. Kimmel, G. Sapiro, "Geodesic active contours", in *Proceedings of the IEEE International Conference on Computer Vision (ICCV'95)*, IEEE Computer Society Press, Cambridge, USA, pp. 694–699, 1995.

[23] V. Caselles, R. Kimmel, G. Sapiro, "Geodesic active contours", *International Journal of Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997.

[24] A. Chakraborty, L.H. Staib, J.S. Duncan, "Deformable Boundary Finding in Medical Images by Integrating Gradient and Region Information", *IEEE Transactions on Medical Imaging*, vol. 15, no. 6, pp. 859–870, 1996.

[25] T.F. Chan & L.A. Vese, "Active contours without edges", *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, February 2001.

[26] S. Chandran, T. Meajima, S. Miyazaki, "Global minima via dynamic programming: Energy minimising active contours", in *Proceedings SPIE Geometric Methods in Computer Vision*, vol. 1570, pp. 391–402, 1991.

[27] C. Chesnaud, P. Refregier, V. Boulet, "Statistical Region Snake-Based Segmentation Adapted to Different Physical Noise Models", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 21, no. 11, pp. 1145–1157, November 1999.

[28] D.L. Chopp & J.A. Sethian, "Motion by Intrinsic Laplacian of Curvature", Tech. Rep., Dept. of Mathematics, Univ. of California, Berkeley, September 1998. CPAM Report PAM-746.

[29] L.D. Cohen, "On active contour models and balloons", *Computer Vision, Graphics, and Image Processing: Image Understanding*, vol. 53, no. 2, pp. 211–218, 1991.

[30] L.D. Cohen, *Variational Methods for Image Processing*, Université Paris Dauphine, May 1995. "Mémoire d'Habilitation à diriger des recherches" Presented together with 10 main publications (in English) during 1988-1995.

[31] L.D. Cohen, "Multiple Contour Finding and Perceptual Grouping using Minimal Paths", *Journal of Mathematical Imaging and Vision*, vol. 14, no. 3, 2001. CEREMADE TR 0101, Jan 2001. To appear.

[32] L.D. Cohen & I. Cohen, "Finite-Element Methods for Active Contour Models and Balloons for 2-D and 3-D Images", *PAMI*, vol. 15, no. 11, pp. 1131–1147, November 1993.

[33] L.D. Cohen & R. Kimmel, "Fast marching the global minimum of active contours", in *Proceedings of the IEEE International Conference on Image Processing (ICIP'96)*, vol. 1, Lausanne, Switzerland,, pp. 473–476, sept 1996.

[34] L.D. Cohen & R. Kimmel, "Global Minimum for Active Contour Models: A Minimal Path approach", *International Journal of Computer Vision*, vol. 24, no. 1, pp. 57–78, Aug. 1997.

[35] M.G. Crandall, H. Ishii, P. L. Lions, "User's guide to viscosity solutions of second order partial linear differential equations", *Bulletin of the American Math. Society*, vol. 27, pp. 1–67, 1992.

[36] O. Cuisenaire, *Distance Transformations: Fast Algorithm and Applications to medical Image Processing*, Ph.D. thesis, Université catholique de Louvain, Belgium, Oct. 1999.

[37] H. Delingette, *Modélisation, déformation et reconnaissance d'objets tridimensionnels à l'aide de maillages simplexes*, Ph.D. thesis, Ecole Centrale Paris, France, July 1994.

[38] H. Delingette, "General Object Reconstruction Based on Simplex Meshes", *International Journal of Computer Vision*, vol. 32, no. 2, pp. 1–36, September 1999.

[39] H. Delingette & J. Montagnat, "New Algorithms for Controlling Active Contours Shape and Topology", in *Proceedings of the Sixth European Conference on Computer Vision (ECCV'00)*, pp. xx–yy, 2000.

[40] H. Delingette & J. Montagnat, "Shape and Topology Constraints on Parametric Active Contours", *CVIU*, vol. 83, no. 2, pp. 140–171, August 2001.

[41] M. Desbrun, *Modeling and Animating highly deformable objects in Computer Graphics.*, Ph.D. thesis, Institut National Polytechnique de Grenoble, December 1997.

[42] T. Deschamps, S.M. Ebeid, L.D. Cohen, "Image Processing Method, System and Apparatus for Processing an Image representing a tubular structure and for constructing a path related to said structure", Patent Pending, March 1999.

[43] E.W. Dijkstra, "A note on two problems in connection with graphs", *Numerische Mathematic*, vol. 1, pp. 269–271, 1959.

[44] L.C. Evans & J. Spruck, "Motion of level-sets by mean curvature", *Journal of Differential Geometry*, vol. 33, pp. 635–681, 1991.

[45] R. Fahrig, A.J. Fox, S. Lownie, D.W. Holdsworth, "Use of a C-Arm System to Generate True Three-dimensional Computed Rotational Angiograms: Preliminary In Vitro and In Vivo Results", *American Journal Neuroradiology*, vol. 18, pp. 1507–1514, 1997.

[46] A.X. Falcao & J.K. Udupa, "Segmentation of 3D objects using live-wire", in *SPIE Medical Imaging*, vol. 3034, Newport Beach, CA, pp. 228–239, February 1997.

[47] A.X. Falcao & J.K. Udupa, "A 3D generalization of user-steered live-wire segmentation", *Medical Image Analysis*, vol. 4, no. 4, pp. 389–402, 2000.

[48] A.X. Falcao, J.K. Udupa, S. Samarasekera, B.E. Hirsch, "User-steered image boundary segmentation", in *SPIE Proceedings*, vol. 2710, pp. 278–288, 1996.

[49] A.X. Falcao, J.K. Udupa, S. Samarasekera, S. Sharma, B.E. Hirsch, R. de A. Lotufo, "User-Steered Image Segmentation Paradigms: Live-Wire and Live-Lane", *Graphical Models and Image Processing*, vol. 60, no. 4, pp. 233–260, 1998.

[50] O.D. Faugeras & R. Keriven, "Variational-Principles, Surface Evolution, PDEs, Level Set Methods, and the Stereo Problem", *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 336–344, March 1998.

[51] L.A. Feldkamp, L.C. Davis, J.W. Kress, "Practical cone-beam algorithm", *Journal of the Optical Society of America*, vol. 1, no. 6, pp. 612–619, June 1984.

[52] C. Fetita, *Analyse morphofonctionnelle des voies aériennes en TDM spiralée volumique*, Ph.D. thesis, Université Paris 5 René Descartes, 2000.

[53] M.A. Fischler, J.M. Tenenbaum, H.C. Wolf, "Detection of roads and linear structures in low-resolution aerial imagery using a multi-source knowledge integration technique", *Computer Graphics and Image Processing*, vol. 15, pp. 201–223, 1981.

[54] R. Florent, "Image processing method and system, and medical examination apparatus for extracting a path following a threadlike structure in an image", Patent Pending, June 1999, 99401348.0.

[55] R. Florent & J. Breitenstein, "Guide-wire extraction in fluoroscopy", Patent Pending, December 1998, 98403323.3.

[56] R. Florent & L. Goubet, "Extraction of Guide-wire with ridgeness + front propagation", Patent Pending, March 2000, 00 401367.8.

[57] A. Frangi, *Three-Dimensional Model-Based Analysis of Vascular and Cardiac images*, Ph.D. thesis, University Medical Center Utrecht, 2001.

[58] A. Frangi, W. Niessen, R.M. Hoogeveen, Th. van Walsum, M.A. Viergever, "Model based quantification of 3D resonance magnetic angiographic images", *IEEE Transactions on Medical Imaging*, vol. 18, no. 10, pp. 946–956, October 1999. Special Issue on Model-based Medical Image Analysis.

[59] A. Frangi, W. Niessen, P.J. Nederkoorn, J. Bakker, W.P.Th.M. Mali, M.A. Viergever, "Quantitative analysis of vessel morphology from 3D MR angiograms:in vitro and in vivo results", *Magnetic Resonance in Medicine*, vol. 45, no. 2, pp. 311–322, February 2001.

[60] A. Frangi, W. Niessen, K.L. Vincken, M.A. Viergever, "Multiscale Vessel Enhancement Filtering", in *Proceedings of the first International Conference on Medical Image Computing and Computer-Assisted Intervention, (MICCAI'98)*, pp. 130–137, 1998.

[61] P. Fua & Y.G. Leclerc, "Model driven edge detection", *Machine Vision and Applications*, vol. 3, pp. 45–56, 1990.

[62] D. Geiger, A. Gupta, L. Costa, J. Vlontzos, "Dynamic programming for detecting, tracking, and matching deformable contours", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 17, no. 3, March 1995.

[63] D. Geman & B. Jedynak, "An Active Testing Model for Tracking Roads in Satellite Images", *PAMI*, vol. 18, no. 1, pp. 1–14, January 1996.

[64] J. Gil & R. Kimmel, "Efficient Dilation, Erosion, Opening and Closing Algorithms", in *International Symposium on Mathematical Morphology and its Applications to Image and Signal Processing V*, Palo-Alto, CA, USA, June 2000.

[65] R. Goldenberg, R. Kimmel, E. Rivlin, M. Rudzsky, "Fast Geodesic Active Contours", in *Scale Space Theories in Computer Vision - Scale Space'99*, pp. 34–45, 1999.

[66] R. Goldenberg, R. Kimmel, E. Rivlin, M. Rudzsky, "Cortex Segmentation: A fast Variational Geometric Approach", in *Workshop on Variational and Level Set Methods in Computer Vision*, Vancouver, Canada, 2001.

[67] R. Goldenberg, R. Kimmel, E. Rivlin, M. Rudzsky, "Fast Geodesic Active Contours", *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1467–1475, October 2001.

[68] J. Gomes & O.D. Faugeras, "Reconciling Distance Functions and Level Sets", in *Scale Space Theories in Computer Vision - Scale Space'99*, pp. 70–81, 1999.

[69] J. Gomes & O.D. Faugeras, "Level Sets and Distance Functions", in *Proceedings of the Sixth European Conference on Computer Vision (ECCV'00)*, 2000.

[70] M. Grayson, "The Heat equation shrinks embedded plane curves to round points", *Journal of Differential Geometry*, vol. 26, pp. 285–314, 1987.

[71] M. Greff, O. Gerard, T. Deschamps, "Adaptation of potential terms in real-time optimal path extraction", Patent Pending, April 2001, FR 01 401 696.8.

[72] G. Guy & G. Medioni, "Inferring Global perceptual contours from local features", *International Journal of Computer Vision*, vol. 20, no. 1/2, pp. 113–133, Oct. 1996.

[73] S. Haker, S. Angenent, A. Tannenbaum, R. Kikinis, "Nondistorting flatening maps and the 3D visualization of colon CT images", *IEEE Transactions on Medical Imaging*, July 2000.

[74] S. Haker, A. Tannenbaum, R. Kikinis, "Mass preserving mappings and image registration", in *Proceedings of the 4th International Conference on Medical Image Computing and Computer-Assisted Intervention, (MICCAI'01)*, W. Niessen & M.A. Viergever (eds.), Springer Verlag, Utrecht, The Netherlands, pp. 120–127, October 2001.

[75] J. Helferty, A. Sherbondy, A. Kiraly, J. Turlington, E.A. Hoffman, G. McLennan, W. Higgins, "Image-guided Endoscopy for Lung-cancer Assessment", in *Proceedings of the IEEE International Conference on Image Processing (ICIP'01)*, pp. 307–310, 2001.

[76] G. Hermosillo, O. Faugeras, J. Gomes, "Unfolding the Cerebral Cortex Using Level Set Methods", in *Scale Space Theories in Computer Vision - Scale Space'99*, pp. 58–69, 1999.

[77] L. Hong, S. Muraki, A. Kaufman, D. Bartz, H. Taosong, "Virtual voyage: interactive navigation in the human colon", in *Proceedings of 24th International Conference on Computer Graphics and Interactive Techniques*, pp. 27–34, August 1997.

[78] S. Jehan-Besson, M. Barlaud, G. Aubert, "Detection and Tracking of Moving Objects Using a New Level Set Based Method", in *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR'00)*, pp. Vol III: 1112–1117, 2000.

[79] S. Jehan-Besson, M. Barlaud, G. Aubert, "Video Objects Segmentation Using Eulerian Region-Based Active Contours", in *Proceedings of the IEEE International Conference on Computer Vision (ICCV'01)*, vol. 1, pp. 353–360, 2001.

[80] F.A. Jolesz, W.E. Loresen, H. Shinmoto, H. Atsumi, S. Nakajima, P. Kavanaugh, P. Saiviroonporn, S.E. Seltzer, S.G. Silverman, M. Phillips, R. Kikinis, "Interactive virtual endoscopy", *American Journal of Radiology*, vol. 169, pp. 1229–1237, 1997.

[81] G.K. Kanizsa, "Subjective Contours", in *Scientific American*, no. 234, 1976.

[82] M. Kass, A. Witkin, D. Terzopoulos, "Snakes: Active contour models", *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[83] R. Kemkers, J. Op de Beek, H. Aerts, R. Koppe, E. Klotz, J.M. Moret, "3D-Rotational Angiography: First clinical application with use of a standard Philips C-arm system", in *Computer Assisted Radiology and Surgery, CARS'98*, 1998.

[84] R. Keriven, *Equations aux Dérivées Partielles, Evolutions de Courbes et de Surfaces et Espaces d'Echelle: Application à la Vision par Ordinateur*, Ph.D. thesis, Ecole Nationale des Ponts et Chaussées, 1997.

[85] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, A. Yezzi, "Gradient flows and geometric active contour models", in *Proceedings of the IEEE International Conference on Computer Vision (ICCV'95)*, IEEE Computer Society Press, Cambridge, USA, pp. 810–815, 1995.

[86] R. Kimmel, A. Amir, A.M. Bruckstein, "Finding shortest paths on surfaces using level sets propagation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 1, pp. 635–640, 1995.

[87] R. Kimmel, N. Kiryati, A.M. Bruckstein, "Distance maps and weighted distance transforms", *Journal of Mathematical Imaging and Vision*, vol. 6, pp. 223–233, May 1996. Special Issue on Topology and Geometry in Computer Vision.

[88] R. Kimmel, N. Kiryati, A.M. Bruckstein, "Multivalued Distance Maps for Motion Planning on Surfaces with Moving Obstacles", *IEEE Trans. Robotics and Automation*, vol. 14, no. 3, pp. 427–436, June 1998.

[89] R. Kimmel & J.A. Sethian, "Fast Marching Methods for Computing Distance Maps and Shortest Paths, CPAM Report 669", Tech. Rep., Center for Pure and Apllied Mathematics, University of California, Berkeley, 1996.

[90] R. Kimmel & J.A. Sethian, "Computing Geodesic Paths on Manifolds", in *Proceedings of the National Academy of Sciences of the United States of America*, vol. 15, University of California, Berkeley, pp. 8431–8435, July 1998.

[91] R. Kimmel & J.A. Sethian, "Optimal algorithm for shape from shading and path planning", in *Mathematical Image Analysis, (MIA'00)*, Paris, September 2000.

[92] R. Kimmel & J.A. Sethian, "Optimal algorithm for shape from shading and path planning", *Journal of Mathematical Imaging and Vision*, vol. 14, no. 3, pp. 237–244, 2001.

[93] N. Kiryati & G. Szekely, "Estimating Shortest Paths and Minimal Distances on Digitized Three-Dimensional Surfaces", *Pattern Recognition*, vol. 26, pp. 1623–1637, 1993.

[94] J.J Koenderink & A.J. van Doorn, "Surface Shape and Curvature Scales", *Image and Vision Computing*, vol. 10, pp. 557–565, 1992.

[95] R. Koppe, E. Klotz, J. Op de Beek, H. Aerts, "3D Vessel reconstruction based in Rotational Angiography", in *Computer Assisted Radiology, CAR'95*, 1995.

[96] R. Kutka & S. Stier, "Extraction of Line Properties Based on Direction Fields", *IEEE Transactions on Medical Imaging*, vol. 15, no. 1, pp. 51–58, 1996.

[97] J.O. Lachaud, *Surface extraction from three-dimensional images: discrete approach and deformable model approach*, Ph.D. thesis, Université Joseph Fourier, july 1998.

[98] J.O. Lachaud & A. Montanvert, "Deformable meshes with automated topology changes for coarse-to-fine three-dimensional surface extraction", *Medical Image Analysis*, vol. 3, no. 2, pp. 187–207, 1999.

[99] S.H. Landis, T. Murray, S. Bolden, P.A. Wingo, "Cancer statistics, 1998", *Cancer Journal for Clinicians*, vol. 48, no. 1, pp. 6–29, 1998.

[100] J.C. Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, Boston, MA, 1991.

[101] O. Lavialle, F. Angella, P. Baylou, "Extension of the Minimal Path Searching for Structure Recovery", in *Proceedings of the IEEE International Conference on Image Processing (ICIP'99)*, Kobe, Japan, pp. 405–406, October 1999.

[102] F. Leymarie & M.D. Levine, "Tracking deformable objects in the plane using an active contour model", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 15, no. 6, pp. 617–634, 1993.

[103] S. Liapis, E. Sifakis, G. Tziritas, "Color And/or Texture Segmentation Using Deterministic Relaxation and Fast Marching Algorithms", in *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR'00)*, vol. 3, pp. 621–624, 2000.

[104] W.E. Lorensen & H.E. Cline, "Marching Cubes: a high resolution 3D surface reconstruction algorithm", *Computer Graphics*, vol. 21, no. 4, pp. 163–169, 1987.

[105] C. Lorenz, I.C. Carlsen, T.M. Buzug, C. Fassnacht, J. Weese, "A Multi-Scale Line Filter with Automatic Scale Selection Based on the Hessian Matrix for Medical Image Segmentation", in *Scale Space Theories in Computer Vision - Scale Space'97*, pp. 152–163, july 1997.

[106] L.M. Lorigo, O.D. Faugeras, W.E.L. Grimson, R. Keriven, R. Kikinis, A. Nabavi, C.F. Westin, "Codimension-Two Geodesic Active Contours for the Segmentation of Tubular Structures", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'00)*, vol. 1, pp. 444–451, 2000.

[107] L.M. Lorigo, O.D. Faugeras, W.E.L. Grimson, R. Keriven, R. Kikinis, A. Nabavi, C.F. Westin, "CURVES: Curve evolution for vessel segmentation", *Medical Image Analysis*, vol. 5, no. 3, pp. 195–206, September 2001.

[108] L.M. Lorigo, O.D. Faugeras, W.E.L. Grimson, R. Keriven, R. Kikinis, C.F. Westin, "Codimension 2 Geodesic Active Contours for MRA Segmentation", in *International Conference on Information Processing in Medical Imaging (IPMI'99)*, Visegrad, Hungary, June 1999.

[109] J.B.A. Maintz, P.A. van den Elsen, M.A. Viergever, "Evaluation of Ridge Seeking Operators for Multimodality Medical Image Matching", *PAMI*, vol. 18, no. 4, pp. 353–365, April 1996.

[110] R. Malladi, R. Kimmel, D. Adalsteinsson, G. Sapiro, V. Caselles, J.A. Sethian, "A Geometric Approach to Segmentation and Analysis of 3D Medical Images", in *Proceedings of mathematical methods in biomedical image analysis, MMBIA'96*, 1996.

[111] R. Malladi & J.A. Sethian, "A Real-Time Algorithm for Medical Shape Recovery", in *Proceedings of the IEEE International Conference on Computer Vision (ICCV'98)*, pp. 304–310, Jan. 1998.

[112] R. Malladi, J.A. Sethian, B.C. Vemuri, "Evolutionary Fronts for Topology-Independent Shape Modeling and Recovery", in *Proceedings of the Third European Conference on Computer Vision (ECCV'94)*, vol. A, Stockholm, Sweden, pp. 3–13, 1994.

[113] R. Malladi, J.A. Sethian, B.C. Vemuri, "Shape Modelling with Front Propagation: A Level Set Approach", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 17, no. 2, pp. 158–175, Feb. 1995.

[114] T. McInerney & D. Terzopoulos, "Topologically Adaptable Snakes", in *Proceedings of the IEEE International Conference on Computer Vision (ICCV'95)*, IEEE Computer Society Press, Cambridge, pp. 840–845, June 1995.

[115] T. McInerney & D. Terzopoulos, "Deformable Models in Medical Image Analysis: A Survey", *Medical Image Analysis*, vol. 1, no. 2, 1996.

[116] T. McInerney & D. Terzopoulos, "T-snakes: Topology adaptive snakes", *Medical Image Analysis*, vol. 4, no. 2, pp. 73–91, 2000.

[117] N. Merlet, J. Zerubia, K.A. Hogda, B. Braathen, K. Heia, "A curvature-dependent energy function for detecting lines in satellite images", in *Proceedings of Scandinavian Conference on Image Analysis*, Tromso, Norway, pp. 699–706, May 1993.

[118] F. Meyer & P. Maragos, "Multiscale Morphological Segmentations based on Watershed, Flooding, and Eikonal PDE", in *Scale Space Theories in Computer Vision - Scale Space'99*, pp. 351–362, 1999.

[119] J.S.B. Mitchell, D.W. Payton, D.M. Keirsey, "Planning and Reasoning for Autonomous Vehicle Control", *International Journal of Intelligent Systems*, vol. 2, pp. 129–198, 1987.

[120] J. Montagnat, *Modéles Déformables pour la Segmentation et la Modélisation d'Images Médicales 3D et 4D*, Ph.D. thesis, Université de Nice Sophia-Antipolis, France, December 1999.

[121] J. Montagnat & H. Delingette, "Globally constrained deformable models for 3D object reconstruction", *IEEE Transactions on Signal Processing*, vol. 71, no. 2, pp. 173–186, December 1998.

[122] U. Montanari, "On the Optimal Detection of Curves in Noisy Pictures", *Communications of the ACM*, vol. 14, no. 5, pp. 335–345, May 1971.

[123] K. Mori, J. Hasegawa, Y. Suenaga, J. Toriwaki, "Automated anatomical labeling of the bronchial branch and its application to the virtual bronchoscopy system", *IEEE Transactions on Medical Imaging*, vol. 19, no. 2, pp. 103–114, February 2000.

[124] K. Mori, J. Hasegawa, J. Toriwaki, H. Anno, K. Katada, "Recognition of Bronchus in Three Dimensional X-Ray CT Images with Applications to Virtualized Bronchoscopy System", in *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR'96)*, pp. 528–532, 1996.

[125] K. Mori, S. Yamazaki, J. Hasegawa, "Extension of virtual bronchoscopy system as a teaching tool", in *Computer Assisted Radiology and Surgery, CARS'99*, pp. 166–170, 1999.

[126] E.N. Mortensen & W.A. Barrett, "Intelligent Scissors for Image Composition", in *Proceedings of Computer Graphics, SIGGRAPH'95*, pp. 191–198, 1995.

[127] E.N. Mortensen & W.A. Barrett, "Interactive Segmentation with Intelligent Scissors", *Graphical Models and Image Processing*, vol. 60, no. 5, pp. 349–384, September 1998.

[128] E.N. Mortensen & W.A. Barrett, "Toboggan-Based Intelligent Scissors with a Four Parameter Edge Model", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99)*, pp. II:452–458, 1999.

[129] E. Mortensen, B. Morse, W. Barrett, J.K. Udupa, "Adaptive boundary detection using live-wire two-dimensional dynamic programming", in *IEEE Proceedings of Computers in Cardiology*, pp. 635–638, October 1992.

[130] M. Naf, O. Kubler, R. Kikinis, M.E. Shenton, G. Szekely, "Characterization and Recognition of 3D Organ Shape in Medical Image Analysis Using Skeletonization", in *Proceedings of mathematical methods in biomedical image analysis, MMBIA'96*, pp. 139–150, 1996.

[131] D. Nain, S. Haker, R. Kikinis, W.E.L. Grimson, "An Interactive Virtual Endoscopy Tool", in *Workshop on Interactive Medical Image Visualization and Analysis*, W. Niessen, S. Olabarriaga, F. Gerritsen (eds.), Utrecht, The Netherlands, pp. 55–60, October 2001.

[132] T. O'Donnell, X.S. Fang, A. Gupta, T.E. Boult, "The Extruded Generalized Cylinder: A Deformable Model for Object Recovery", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pp. 174–181, 1994.

[133] S.D. Olabarriaga, *Human-Computer Interaction for the Segmentation of Medical Images*, Ph.D. thesis, University Medical Center Utrecht, December 1999.

[134] S.D. Olabarriaga & A.W.M. Smeulders, "Interaction in the segmentation of medical images: A survey", *Medical Image Analysis*, vol. 5, no. 2, pp. 127–142, June 2001.

[135] S. Osher & J.A. Sethian, "Fronts propagating with curvature dependent speed: algorithms based on the Hamilton-Jacobi formulation", *Journal of Computational Physics*, vol. 79, pp. 12–49, 1988.

[136] D.S. Paik, C.F. Beaulieu, R.B. Jeffrey, G.D. Rubin, S. Napel, “Automated flight path planning for virtual endoscopy”, *Medical Physics*, vol. 25, no. 5, pp. 629–637, 1998.

[137] N. Paragios, *Geodesic Active Regions and Level Set Methods: Contributions and Applications in Artificial Vision*, Ph.D. thesis, Université de Nice Sophia-Antipolis, France, 2000.

[138] N. Paragios & R. Deriche, “Geodesic Active Regions for Motion Estimation and Tracking”, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV’99)*, Corfou, Greece, pp. 688–694, September 1999.

[139] N. Paragios & R. Deriche, “Geodesic Active Regions for Supervised Texture Segmentation”, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV’99)*, Corfou, Greece, September 1999.

[140] N. Paragios, O. Mellina-Gotardo, V. Ramesh, “Gradient Vector Flow Fast Geodesic Active Contours”, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV’01)*, vol. 1, Vancouver, Canada, pp. 67–73, 2001.

[141] G. Picinbono, “Modèle géométrique de contour déformable implicite pour la segmentation et la simulation”, Master’s thesis, Université de Nice-Sophia Antipolis, 1997.

[142] L. Piegl & W. Tiller, *The NURBS Book*, Springer Verlag, Berlin, 1996.

[143] C. Pisupati, L. Wolff, W. Mitzner, E. Zerhouni, “A Central Axis Algorithm for 3D Bronchial Tree Structures”, in *SCV95*, pp. 259–264, 1995.

[144] F. Preteux, C. Fetita, P. Grenier, A. Capderou, “Modeling, segmentation and caliber estimation of bronchi in high-resolution computerized tomogrpahy”, *Journal of Electronic Imaging*, vol. 8, no. 1, pp. 36–45, 1999.

[145] L.R. Rabiner, A.E. Rosenberg, S.E. Levinson, “Considerations In Dynamic Time Warping Algorithms For Discrete Word Recognition”, *IEEE Transactions On Acoustics, Speech and Signal Processing*, vol. 26, no. 6, pp. 575–582, December 1978.

[146] P. Rogalla, J.T. van Schletinga, B. Hamm (eds.), *Virtual Endoscopy and Related 3D Techniques*, Diagnostic Imaging, Medical Radiology, Springer Verlag, Berlin, 2001.

[147] P. Rogalla, B. Verdonck, R. Truyen, B. Hamm, “Efficacy of automatic path tracking for virtual colonoscopy”, in *Radiological Society of North America (RSNA)*, 1999.

[148] R. Ronfard, “Region-Based Strategies for Active Contour Models”, *International Journal of Computer Vision*, vol. 13, no. 2, pp. 229–251, October 1994.

[149] N. Rougon & F. Preteux, “Generalized Geodesic Active Contours”, in *Reconnaissance de Formes et Intelligence Artificielle (RFIA’98)*, Clermond-Ferrant, pp. 287–296, January 1998.

[150] E. Rouy & A. Tourin, “A Viscosity Solution Approach to Shape-From-Shading”, *SIAM Journal of Numerical Analysis*, vol. 29, pp. 867–884, 1992.

[151] D. Rutovitz, “Data structures for operations on digital images”, *Pictorial pattern recognition*, pp. 105–133, 1968.

[152] C. Samson, L. Blanc-Feraud, G. Aubert, J. Zerubia, “A Variational Model for Image Classification and Restoration”, *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 22, no. 5, pp. 460–472, May 2000.

[153] G. Sapiro, “Vector-valued active contours”, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’96)*, San Francisco, USA, 1996.

[154] A. Sarti, R. Malladi, J.A. Sethian, “Subjective surfaces: A method for completing missing boundaries”, in *Proceedings of the National Academy of Sciences of the United States of America*, vol. 12, pp. 6258–6263, 2000.

[155] R. Sedgewick, *Algorithms in C - Parts 1-4*, 3 ed., Addison-Wesley, Reading, Massachusetts, USA, 1998.

[156] J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, New York, 1982.

[157] J. Serra, “Morphological Descriptions Using Three-dimensional Wavefronts”, Tech. Rep. [N -30/01/MM], Centre de Morphologie Mathématique, Ecole des Mines de Paris, June 2001.

[158] J.A. Sethian, “A review of recent numerical algorithms for hypersurfaces moving with curvature dependent flows”, *Journal of Differential Geometry*, vol. 31, pp. 131–161, 1989.

[159] J.A. Sethian, “Curvature flow and entropy conditions applied to grid generation”, *Journal of Computational Physics*, vol. 115, pp. 440–454, 1994.

[160] J.A. Sethian, “A review of the theory, algorithms, and applications of level set methods for propagating interfaces”, *Acta Numerica*, 1995.

[161] J.A. Sethian, “A fast marching level set method for monotonically advancing fronts”, *Proceedings of the Natural Academy of Sciences*, vol. 93, no. 4, pp. 1591–1595, Feb. 1996.

[162] J.A. Sethian, “Tracking Interfaces with Level Sets”, *American Scientist*, May-June 1997.

[163] J.A. Sethian, *Level set methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Sciences*, 2nd ed., Cambridge University Press, University of California, Berkeley, 1999.

[164] A. Shaashua & S. Ullman, “Structural saliency: The detection of globally salient structures using a locally connected network”, in *Proc. Second IEEE International Conference on Computer Vision (ICCV’88)*, pp. 321–327, December 1988.

[165] J. Shah, “A common framework for curve evolution, segmentation and anisotropic diffusion”, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’96)*, San Francisco, USA, 1996.

[166] P. Shirley & A. Tuchman, “A polygonal approximation to direct scalar volume rendering”, *Computer Graphics*, vol. 24, no. 5, pp. 63–70, 1990.

[167] E. Sifakis, C. Garcia, G. Tziritas, “Bayesian level sets for image segmentation”, in *Visual Communication and Image Representation*, 2001.

[168] American Cancer Society (ed.), *Cancer Facts and Figures*, 2, ACS, Atlanta, 1999.

[169] M. Sonka, W. Park, E. Hoffman, “Rule-Based Detection of Intrathoracic airway trees”, *IEEE Transactions on Medical Imaging*, vol. 15, no. 3, pp. 314–326, June 1996.

[170] M. Sussman, P. Smereka, S.J. Osher, “A level set method for computing solutions to incompressible two-phase flow”, *Journal of Computational Physics*, vol. 114, pp. 146–159, 1994.

[171] R. Szeliski, D. Tonnesen, D. Terzopoulos, “Modeling Surfaces of Arbitrary Topology with Dynamic Particles”, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’93)*, pp. 82–87, 1993.

[172] H. Tek & B. Kimia, “Image Segmentation by Reaction-Diffusion Bubbles”, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV’95)*, Cambridge, USA, pp. 156–162, June 1995.

[173] H. Tek & B. Kimia, “Boundary smoothing via symmetry transforms”, *to appear in Journal of Mathematical Imaging and Vision, Special issue on Mathematics and Image Analysis MIA’00*, 2001.

[174] D. Terzopoulos, “On matching deformable models to images: Direct and iterative solutions”, *Topical meeting on machine vision, Technical Digest Series, Optical Society of America*, vol. 12, pp. 160–167, 1987.

[175] D. Terzopoulos, A.P. Witkin, M. Kass, “Symmetry-Seeking Models and 3D Object Reconstruction”, *International Journal of Computer Vision*, vol. 1, no. 3, pp. 211–221, October 1987.

[176] E. Thiel & A. Montanvert, “Chamfer masks: discrete distance functions, geometrical properties and optimization”, in *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR’92)*, The Hague, Netherlands, pp. 244–247, 1992.

[177] A. Vasilevskiy & K. Siddiqi, “Flux Maximizing Geometric Flows”, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV’01)*, vol. 1, Vancouver, Canada, pp. 149–154, 2001.

[178] P.W. Verbeek & B.J.H. Verwer, “Shading from shape, the eikonal equation solved by gray-weighted distance transform”, *Pattern Recognition Letters*, vol. 11, pp. 681–690, 1990.

[179] A. Verroust & F. Lazarus, "Extracting skeletal curves from 3D scattered data", Tech. Rep., Unité de Recherche INRIA Rocquencourt, 1997. RR 3250.

[180] L. Vincent, *Algorithmes Morphologiques à base de Files d'Attente et de Lacets - Extension aux graphes*, Ph.D. thesis, Ecole des Mines de Paris, Xerox Imaging, Palo Alto, 1991.

[181] L. Vincent & P. Soille, "Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 13, no. 6, pp. 583–598, June 1991.

[182] J. Weickert, B.M. ter Haar Romeny, M.A. Viergever, "Efficient and Reliable Schemes for Nonlinear Diffusion Filtering", *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 398–410, March 1998.

[183] R. Whitaker, "Algorithms for implicit deformable models", in *Proceedings of the IEEE International Conference on Computer Vision (ICCV'95)*, IEEE Computer Society Press, Cambridge, USA, pp. 822–827, 1995.

[184] R.T. Whitaker, "A Level-Set Approach to 3D Reconstruction from Range Data", *International Journal of Computer Vision*, vol. 29, no. 3, pp. 203–231, 1998.

[185] R.T. Whitaker & D.E. Breen, "Level-Set Models for the Deformation of Solid Objects", in *Proceedings of Implicit Surfaces, Eurographics/Siggraph (IS'98)*, pp. 19–35, June 1998.

[186] S. Wildermuth & C.H. Stern, "Interactive Definition of Endoluminal Aortic Stent Size and Morphology Based on Virtual Angioscopic Rendering of 3D MRA", in *Radiological Society of North America (RSNA)*, p. 226, 1998.

[187] L.R. Williams & D.W. Jacobs, "stochastic completion fields: a neural model of illusory contour shape and salience", in *Proceedings of the IEEE International Conference on Computer Vision (ICCV'95)*, Cambridge, USA, pp. 408–415, June 1995.

[188] A. Witkin & P. Heckbert, "Using particles to sample and control implicit surfaces", in *Proceedings of the International Conference on Computer Graphics (SIGGRAPH'94)*, pp. 269–278, 1994.

[189] C.Y. Xu & J.L. Prince, "Gradient Vector Flow: A New External Force for Snakes", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pp. 66–71, 1997.

[190] C.Y. Xu & J.L. Prince, "Generalized gradient vector flow external forces for active contours", *IEEE Transactions on Signal Processing*, vol. 71, no. 2, pp. 131–139, December 1998.

[191] C.Y. Xu & J.L. Prince, "Snakes, Shapes, and Gradient Vector Flow", *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 359–369, March 1998.

[192] G. Yeorong, D.R. Stelts, W. Jie, D.J. Vining, "Computing the centerline of a colon: a robust and efficient method based on 3D skeletons", *Journal of Computer-Assisted Tomography*, vol. 23, no. 5, pp. 786–794, 1999.

[193] A. Yezzi, A. Tsai, A. Willsky, "Medical image segmentation via coupled curve evolution equations with global constraints", in *Proceedings of mathematical methods in biomedical image analysis, MMBIA'00*, Hilton Head Island, USA, pp. 12–19, June 2000.

[194] X. Zeng, L.H. Staib, R.T. Schultz, J.S. Duncan, "Volumetric Layer Segmentation Using Coupled Surfaces Propagation", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'98)*, pp. 708–715, 1998.

[195] X. Zeng, L.H. Staib, R.T. Schultz, J.S. Duncan, "Segmentation and Measurement of the Cortex from 3-D MR Images Using Coupled-Surfaces Propogation", *IEEE Transactions on Medical Imaging*, vol. 18, no. 10, pp. 927–937, October 1999.

[196] S.C. Zhu & A. Yuille, "Region Competition: Unifying Snakes, Region Growing, and Bayes/MDL for Multiband Image Segmentation", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 18, no. 9, pp. 884–900, September 1996.

Publications

Publications in International Journals:

- T. Deschamps, L. D. Cohen, "Fast extraction of minimal paths in 3D images and applications to virtual endoscopy", *Medical Image Analysis*, Vol. 5, No. 4 , August 2001.

Publications in International Conference Proceedings:

- T. Deschamps and L. D. Cohen, "Grouping connected components using minimal path techniques", *Computer Vision and Pattern Recognition (CVPR'01)*, Kauai, Hawai, December 2001
- R. Truyen, T. Deschamps and L. D. Cohen, "Clinical evaluation of an automatic path tracker for virtual colonoscopy." *Medical Image Computing and Computer-Assisted Intervention (MICCAI'01)*, Utrecht, Netherlands, October 2001.
- T. Deschamps, L. D. Cohen, "Multiple path extraction and perceptual grouping as a set of energy minimizing paths." *Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR'01)*, Sophia-Antipolis, France, September 2001.
- R. Truyen, P. Lefere, S. Gryspeerdt and T. Deschamps, "Efficacy of automatic path tracking in virtual colonoscopy", *Computer Assisted Radiology and Surgery (CARS'01)*, Berlin, June 2001.
- M. Benetiere, V. Bottreau, A. Collet-Billon and T. Deschamps, "Scalable Compression of 3D Medical Datasets using a (2D+t) Wavelet Video Coding Scheme." *Sixth International Symposium on Signal Processing and its Applications (ISSPA'01)*, Kuala Lumpur, Malaysia, August 2001.
- R. Truyen, P. Lefere, S. Gryspeerdt and T. Deschamps, "Speed and robustness of (semi-)automatic path tracking", *Virtual Colonoscopy Symposium*, Boston, October 2000.
- T. Deschamps, L. D. Cohen, "Minimal Paths in 3D Images and Application to Virtual Endoscopy", *European Conference on Computer Vision, ECCV'00* (held in Dublin, Ireland, June 2000).
- T. Deschamps, L. D. Cohen, "Path extraction in 3D medical images for virtual endoscopy", *Computer Aided Surgery, Medical Robotics, and Medical Imaging, ISRA-CAS'00* (held in Haifa, Israel, May 2000).
- T. Deschamps, J.M. Létang, L. D. Cohen and B. Verdonck, "Automatic construction of minimal paths in 3D images for virtual endoscopy", in *Computer Assisted Radiology and Surgery, CARS'99* (held in Paris, France, June 1999).

Patents:

- T. Deschamps and S. M. Ebeid and L. D. Cohen "Image Processing Method, System and Apparatus for Processing an Image representing a tubular structure and for constructing a path related to said structure", (march 1999).
- M. Greff and O. Gerard and T. Deschamps "Adaptation of potential terms in real-time optimal path extraction", (April 2001).

Curriculum Vitae

The author was born in Paris, France, on January 19, 1974. After having received general secondary education *(Collège Bartholdi, Boulogne)*, he went to an intermediate school *(Lycée Jacques Prévert, Boulogne)* from which he obtained his "baccalauréat" in 1992. During two years, he endured the preparatory classes for French Engineering schools *(Lycée Claude Bernard, Paris)*, and decided, instead of continuing a third year, to enter university.

In June 1997, he received a Bachelor of Science degree in Mathematics and Computer Science from Paris Dauphine University. And in September 1998, he received a Master of Science with honors in Mathematics, Image Processing and Artificial Intelligence, at CMLA laboratory, Ecole Normale Superieure, Cachan. His graduation project concerned the development of a path tracker for virtual endoscopy in 3D medical image data by means of shortest path techniques and was carried out at the Medical Imaging Systems (MediSys) group of Philips Research France(PRF), in cooperation with EasyVision Advanced Development group of Philips Medical Systems (Best, the Netherlands).

In the subsequent month, he started as a Ph.D. student at the CEREMADE Laboratory, University Paris Dauphine, on a research project concerning the extraction of paths and surfaces in medical imaging using level-sets framework. The project was carried out in the MediSYs Department of PRF (Suresnes, France). The results are described in this thesis.

He was awarded a post-doctoral fellowship, for research on electron microscopy and confocal microscope imaging, to be carried out at the Computing Sciences Division in cooperation with the Life Science Division, Lawrence Berkeley National Laboratory (Berkeley, CA, United States). This project will start in January 2002.

Extraction de Courbes et Surfaces par Méthodes de Chemins Minimaux et Ensembles de Niveaux. Applications en Imagerie Medicale 3D.

Thomas Deschamps
Medisys - Philips Recherche France, B.P. 301, 92156 Suresnes Cédex, France.

Dans cette thèse nous nous interessons à l'utilisation des méthodes de chemins minimaux et des méthodes de contours actifs par Ensembles de Niveaux, pour l'extraction de courbes et de surfaces dans des images médicales 3D.

Dans un premier temps, nous nous sommes attachés à proposer un eventail varié de techniques d'extraction de chemins minimaux dans des images 2D et 3D, basées sur la résolution de l'équation Eikonal par l'algorithme du Fast Marching. Nous avons montré des résultats de ces techniques appliquées à des problèmes d'imagerie médicale concrets, notamment en construction de trajectoires 3D pour l'endoscopie virtuelle, et en segmentation interactive, avec possibilité d'apprentissage.

Dans un deuxième temps, nous nous sommes interessés à l'extraction de surfaces. Nous avons developpé un algorithme rapide de pré-segmentation, sur la base du formalisme des chemins minimaux. Nous avons étudié en détail la mise en place d'une collaboration entre cette méthode et celle des Ensembles de Niveaux, dont un des avantages communs est de ne pas avoir d'à priori sur la topologie de l'objet à segmenter. Cette méthode collaborative a ensuite été testée sur des problèmes de segmentation et de visualisation de pathologies telles que les anévrismes cérébraux et les polypes du colon.

Dans un troisième temps nous avons fusionné les résultats des deux premieres parties pour obtenir l'extraction de surfaces, et des squelettes d'objets anatomiques tubulaires. Les squelettes des surfaces fournissent des trajectoires que nous utilisons pour déplacer des cameras virtuelles, et nous servent à définir les sections des objets lorsque nous voulons mesurer l'étendue d'une pathologie. La dernière partie regroupe des applications de ces méthodes à l'extraction de structures arborescentes. Nous étudions le cas des arbres vasculaires dans des images médicales 3D de produit de contraste, ainsi que le problème plus difficile de l'extraction de l'arbre bronchique sur des images scanners des poumons.

Mots clés : *Chemins minimaux, modèles déformables implicites, segmentation, imagerie médicale 3D, méthodes variationnelles, Level-Sets, Fast-Marching.*

Curve and Shape Extraction with Minimal Path and Level-Sets techniques. Applications to 3D Medical Imaging.

In this thesis, we focus on the use of minimal path techniques and Level-Sets active contours, for curve and shape extraction in 3D medical images.

In the first part of thesis, we worked upon the reduction of the computing cost for path extraction. We proposed several path extraction algorithms for 2D as well as for 3D images. And we applied those techniques to real medical imaging problems, in particular automatic path extraction for virtual endoscopy and interactive and real-time path extraction with on-the-fly training.

In the second part, we focused on surface extraction. We developed a fast algorithm for pre-segmentation, on the basis of the minimal path formalism of the first part. We designed a collaborative method between this algorithm and a Level-Sets formulation of the problem, which advantage is to be able to handle any topological change of the surfaces segmented. This method was tested on different segmentation problems, such as brain aneurysms and colon polyps, where target is accuracy of the segmentation, and enhanced visualization of the pathologies.

In the last part of the thesis, we mixed results from previous part to design a specific method for tubular shape description and segmentation, where description is the extraction of the underlying skeleton of our objects.

The skeletons are trajectories inside our objects, which are used as well for virtual inspection of pathologies, as for accurate definition of cross-sections of our tubular objects. In the last chapter we show applications of our algorithms to the extraction of branching structures. We study the vascular tree extraction in contrast enhanced medical images, and we apply the same principle to the more complex problem of the bronchial tree extraction in multi-slice CT scanners of the lungs.

Keywords: *Minimal Paths, implicit deformable models, segmentation, 3D medical imaging, variational methods, Level-Sets, Fast-Marching.*